A Black-Box Approach to Post-Quantum Zero-Knowledge in Constant Rounds

Nai-Hui Chia[1], Kai-Min Chung[2], and Takashi Yamakawa[3]

[1]QuICS, University of Maryland nchia@umd.edu
[2]Institute of Information Science, Academia Sinica kmchung@iis.sinica.edu.tw
[3]NTT Secure Platform Laboratories takashi.yamakawa.ga@hco.ntt.co.jp

October 30, 2023

Abstract

In a recent seminal work, Bitansky and Shmueli (STOC '20) gave the first construction of a constant round zero-knowledge argument for **NP** secure against quantum attacks. However, their construction has several drawbacks compared to the classical counterparts. Specifically, their construction only achieves computational soundness, requires strong assumptions of quantum hardness of learning with errors (QLWE assumption) and the existence of quantum fully homomorphic encryption (QFHE), and relies on non-black-box simulation.

In this paper, we resolve these issues at the cost of weakening the notion of zero-knowledge to what is called ϵ-zero-knowledge. Concretely, we construct the following protocols:

- We construct a constant round interactive proof for **NP** that satisfies *statistical* soundness and *black-box* ϵ-zero-knowledge against quantum attacks assuming the existence of *collapsing hash functions*, which is a quantum counterpart of collision-resistant hash functions. Interestingly, this construction is just an adapted version of the classical protocol by Goldreich and Kahan (JoC '96) though the proof of ϵ-zero-knowledge property against quantum adversaries requires novel ideas.
- We construct a constant round interactive argument for **NP** that satisfies computational soundness and *black-box* ϵ-zero-knowledge against quantum attacks only assuming the existence of post-quantum one-way functions.

At the heart of our results is a new quantum rewinding technique that enables a simulator to extract a committed message of a malicious verifier while simulating verifier's internal state in an appropriate sense.

1 Introduction

Zero-Knowledge Proof. Zero-knowledge (ZK) proof [GMR89] is a fundamental cryptographic primitive, which enables a prover to convince a verifier of a statement without giving any additional "knowledge" beyond that the statement is true. In the classical setting, there have been many feasibility results on ZK proofs for specific languages including quadratic residuosity [GMR89], graph isomorphism [GMW91], statistical difference problem [SV03] etc., and for all **NP** languages assuming the existence of one-way functions (OWFs) [GMW91, Blu86]. On the other hand, van de Graaf [Gra97] pointed out that there is a technical difficulty to prove security of these protocols against quantum attacks. Roughly, the difficulty comes from the fact that security proofs of these results are based on a technique called *rewinding*, which cannot be done when an adversary is quantum due to the no-cloning theorem. Watrous [Wat09] considered *post-quantum ZK proof*, which means a classical interactive proof that satisfies (computational) zero-knowledge property against quantum malicious verifiers, and showed that some of the classical constructions above are also post-quantum ZK. Especially, he introduced a new *quantum rewinding technique* which is also applicable to quantum adversaries and proved that 3-coloring protocol of Goldreich, Micali, and Wigderson [GMW91] is secure against quantum attacks assuming that the underlying OWF is post-quantum secure, i.e., uninvertible in quantum polynomial-time (QPT).[1] Since the 3-coloring problem is **NP**-complete, this means that there exists a post-quantum ZK proof for all **NP** languages assuming the existence of post-quantum OWFs.

Round Complexity. An important complexity measure of ZK proofs is *round complexity*, which is the number of interactions between a prover and verifier. In this aspect, the 3-coloring protocol [GMW91] (and its quantumly secure version [Wat09]) is not satisfactory since that requires super-constant number of rounds.[2] Goldreich and Kahan [GK96] gave the first construction of a constant round ZK proof for **NP** assuming the existence of collision-resistant hash function in the classical setting. However, Watrous' rewinding technique does not seem to work for this construction (as explained in Sec. 1.2), and it has been unknown if their protocol is secure against quantum attacks.

Recently, Bitansky and Shmueli [BS20] gave the first construction of post-quantum ZK *argument* [BC90] for **NP**, which is a weakened version of post-quantum ZK proof where soundness holds only against computationally bounded adversaries. In addition to weakening soundness to computational one, there are several drawbacks compared to classical counterparts. First, they assume strong assumptions of quantum hardness of learning with erros (QLWE assumption) [Reg09] and the existence of quantum fully homomorphic encryption (QFHE) [Mah18a, Bra18]. Though the QLWE assumption is considered fairly standard due to reductions to worst-case lattice problems [Reg09, Pei09, BLP+13], a construction of QFHE requires circular security of an QLWE-based encryption scheme, which has no theoretical evidence. In contrast, a constant round *classical* ZK argument for **NP** is known to exist under the minimal assumption of the existence of OWFs [FS90, PW09]. Second, their security proof of quantum ZK property relies on a novel *non-black-box* simulation technique, which makes use of the actual description of malicious verifier instead of using it as a black-box. In contrast, classical counterparts can be obtained by black-box simulation [FS90, GK96,

[1] Strictly speaking, Watrous' assumption is a statistically binding and post-quantum computationally hiding commitment scheme, and he did not claim that this can be constructed under the existence of post-quantum OWFs. However, we can see that such a commitment scheme can be obtained by instantiating the construction of [Nao91, HILL99] with a post-quantum OWF.

[2] 3-round suffices for achieving a constant soundness error, but super-constant times sequential repetitions are needed for achieving negligible soundness error (i.e., a cheating prover can let a verifier accept on a false statement only with a negligible probability). Negligible soundness error is a default requirement in this paper.

PW09]. Therefore, it is of theoretical interest to ask if we can achieve constant round quantum ZK by black-box simulation. Third, somewhat related to the second issue, their construction also uses building blocks in a non-black-box manner, which makes the actual efficiency of the protocol far from practical. Again, classical counterparts are known based on black-box constructions [GK96, PW09].

Given the state of affairs, it is natural to ask the following questions:

1. Are there constant round post-quantum ZK proofs for **NP** instead of arguments?
2. Are there constant round post-quantum ZK proofs/arguments for **NP** from weaker assumptions than those in [BS20]?
3. Are there constant round post-quantum ZK proofs/arguments for **NP** based on black-box simulation and/or black-box construction?
4. Are known constructions of constant round classical ZK proofs/arguments for **NP** (e.g., [FS90, GK96, PW09]) secure against quantum attacks if we instantiate them with post-quantum building blocks?

1.1 Our Results

In this work, we partially answer the above questions affirmatively at the cost of weakening the quantum ZK property to *quantum ϵ-ZK*, which is the quantum version of ϵ-ZK introduced in [DNS04].[3]

Quantum ϵ-Zero-Knowledge. The standard quantum ZK property roughly requires that for any QPT V^*, there exists a QPT simulator $\mathcal{S}$ that simulates the interaction between V^* and an honest prover so that the simulation is indistinguishable from the real execution against any QPT distinguishers. On the other hand, in quantum ϵ-ZK, a simulator is allowed to depend on a "accuracy parameter" ϵ. That is, it requires that for any QPT malicious verifier V^* and a noticeable accuracy parameter ϵ, there exists a QPT simulator $\mathcal{S}$ *whose running time polynomially depends on* ϵ^{-1} that simulates the interaction between V^* and an honest prover so that no QPT distinguisher can distinguish it from real execution with advantage larger than ϵ. Though this is a significant relaxation of quantum ZK, this still captures meaningful security. For example, we can see that quantum ϵ-ZK implies both quantum versions of witness indistinguishability and witness hiding similarly to the analogous claims in the classical setting [BKP19].[4] Moreover, by extending the observation in [DNS04] to the quantum setting, we can see the following: Suppose that a QPT malicious verifier solves some puzzle whose solution is efficiently checkable (e.g., finding a witness of an **NP** statement) after an interaction between an honest prover. Then, quantum ϵ-ZK implies that if the verifier succeeds in solving the puzzle with noticeable probability p after the interaction, then there is a QPT algorithm (whose running time polynomially depends on p^{-1}) that solves the same puzzle with noticeable probability (say, $p/2$) *without interacting with the honest prover*. This captures the naive intuition of the ZK property that "anything that can be done after the execution can be done without execution" in some sense, and this would be sufficient in many cryptographic applications. Thus we believe that quantum ϵ-ZK is conceptually a similar notion to the standard quantum ZK. More discussion on (quantum) ϵ-ZK and other related notions of ZK can be found in Sec. 1.3.

[3] ϵ-ZK was originally called ϵ-knowledge, but some later works [BKP18, FGJ18] call it ϵ-ZK. We use ϵ-ZK to clarify that this is a variant of ZK.

[4] Actually, [BKP19] shows that even weaker notion called *weak ZK* suffices for witness indistinguishability and witness hiding. See also Sec. 1.3.

Our Constructions. We give two constructions of constant round quantum ϵ-ZK protocols.

- We construct a constant round quantum ϵ-ZK *proof* for **NP** assuming the existence of *collapsing hash functions* [Unr16b, Unr16a], which is considered as a counterpart of collision-resistant hash functions in the quantum setting. Especially, we can instantiate the construction based on the QLWE assumption. Our construction is fully black-box in the sense that both simulation and construction rely on black-box usage of building blocks and a malicious verifier. Interestingly, this construction is just an adapted version of the classical protocol of [GK96] though the proof of quantum ϵ-zero-knowledge property requires novel ideas.
- We construct a constant round quantum ϵ-ZK argument for **NP** assuming the minimal assumption of the existence of *post-quantum OWFs*. This construction relies on black-box simulation, but the construction itself is non-black-box.

At the heart of our results is a new quantum rewinding technique that enables a simulator to extract a committed message of a malicious verifier while simulating verifier's internal state in some sense. We formalize this technique as an *extraction lemma*, which we believe is of independent interest.

1.2 Technical Overview

Though we prove a general lemma which we call extraction lemma (Lemma 4.2) and then prove quantum ϵ-ZK of our constructions based on that in the main body, we directly explain the proof of quantum ϵ-ZK without going through such an abstraction in this overview.

Known Classical Technique and Difficulty in Quantum Setting. First, we review a classical constant round ZK proof by Goldreich and Kahan [GK96] (referred to as GK protocol in the following), and explain why it is difficult to prove quantum ZK for this protocol by known techniques. GK protocol is based on a special type of 3-round proof system called Σ-protocol.[5] In a Σ-protocol, a prover sends the first message a, a verifier sends the second message e referred to as a *challenge*, which is just a public randomness, and the prover sends the third message z. A Σ-protocol satisfies a special type of honest-verifier ZK, which ensures that if a challenge e is fixed, then one can simulate the transcript (a, e, z) without using a witness. Though this may sound like almost the standard ZK property, a difficulty when proving ZK is that a malicious verifier may *adaptively* choose e depending on a, and thus we cannot fix e at the beginning. To resolve this issue, the idea of GK protocol is to let the verifier commit to a challenge e at the beginning of the protocol. That is, GK protocol roughly proceeds as follows:[6]

1. A verifier sends a commitment com to a challenge e of a Σ-protocol.
2. The prover sends the first message a of the Σ-protocol.
3. The verifier opens com to open a challenge e and its opening information r (i.e., the randomness used for the commitment).
4. The prover aborts if the verifier's opening is invalid. Otherwise it sends the third message z of the Σ-protocol.

[5] In this paper, we use Σ-protocol to mean a parallel repetition version where soundness error is reduced to negligible.

[6] We note that this construction is based on an earlier work of [BCY91].

When proving the ZK property of GK protocol, they rely on a *rewinding* argument. That is, a simulator first runs the protocol with a malicious verifier until Step 3 to extract a committed message e inside com, and then rewind the verifier's state back to just after Step 1, and then simulates the transcript by using the extracted knowledge of e.

On the other hand, this strategy does not work if we consider a quantum malicious verifier since a quantum malicious verifier may perform measurements in Step 3, which is in general not reversible. In other words, since we cannot copy the verifier's internal state after Step 1 due to the no-cloning theorem, we cannot recover that state after running the protocol until Step 3.

Watrous [Wat09] proved that we can apply a rewinding argument for quantum verifiers under a certain condition. Roughly speaking, the condition is that there is a simulator that succeeds in simulation for quantum verifiers with a fixed (verifier-independent) and noticeable probability. For example, if the challenge space is polynomial size, then a simulator that simply guesses a challenge e suffices. However, for achieving negligible soundness error, the challenge space should be super-polynomial size, in which case it seems difficult to construct such a simulator. Also, relaxing quantum ZK to quantum ϵ-ZK does not seem to resolve the issue in any obvious way.

1.2.1 Quantum Analysis of GK Protocol.

In spite of the above mentioned difficulty, we succeed in proving quantum ϵ-ZK for a slight variant of GK protocol. In the following, we explain the idea for our results.

Simplified Goal: Simulation of Non-Aborting Case. First, we apply a general trick introduced in [BS20], which simplifies the task of proving quantum ZK. In GK protocol, we say that a verifier aborts if it fails to provide a valid opening to com in Step 3. Then, for proving quantum ZK of the protocol, it suffices to construct two simulators $\mathsf{Sim}_{\mathsf{a}}$ and $\mathsf{Sim}_{\mathsf{na}}$ that work only when the verifier aborts and does not abort and they do not change the probability that the verifier aborts too much, respectively. The reason is that if we randomly choose either of these two simulators and just run the chosen one, then the simulation succeeds with probability $1/2$ since the guess of if the verifier aborts is correct with probability $1/2$. Then, we can apply Watrous' rewinding technique to convert it to a full-fledged simulator. Essentially the same trick also works for quantum ϵ-ZK.

Moreover, it is easy to construct $\mathsf{Sim}_{\mathsf{a}}$ because the first message of a Σ-protocol can be simulated without witness, and one need not provide the third message to the verifier when it aborts. Therefore, the problem boils down to constructing a simulator $\mathsf{Sim}_{\mathsf{na}}$ that works only when the verifier does not abort.

Initial Observations. For explaining how to construct $\mathsf{Sim}_{\mathsf{na}}$, we start by considering the simplest case where a verifier never aborts. Moreover, suppose that the commitment scheme used for committing to a challenge e satisfies the strict-binding property [Unr12], i.e., for any commitment com, there is at most one valid message and randomness. Then, a rewinding strategy similar to the classical case works since, in this case, the verifier's message in Step 3 is information-theoretically determined, and such a deterministic computation does not collapse a quantum state in general.[7] However, for ensuring statistical soundness, we have to use a statistically hiding commitment, which cannot be strict-binding. Fortunately, this problem can be resolved by using *collapse-binding* commitments [Unr16b], which roughly behave similarly to strict-binding commitments for any *com-*

[7] This is also observed in [BS20].

putationally bounded adversaries.[8] Since this is rather a standard technique, in the rest of this overview, we treat the commitment as if it satisfies the strict-binding property.

Next, we consider another toy example where a verifier sometimes aborts. Suppose that a malicious verifier V^* is given an initial state $\frac{1}{\sqrt{2}}(|\psi_{\mathsf{a}}\rangle + |\psi_{\mathsf{na}}\rangle)$ in its internal register $\mathbf{V}$ where $|\psi_{\mathsf{a}}\rangle$ and $|\psi_{\mathsf{na}}\rangle$ are orthogonal, and runs as follows:

1. V^* randomly picks e, honestly generates a commitment com to e, and sends it to the prover (just ignoring the initial state).
2. After receiving a, V^* performs a projective measurement $\{|\psi_{\mathsf{a}}\rangle\langle\psi_{\mathsf{a}}|, I - |\psi_{\mathsf{a}}\rangle\langle\psi_{\mathsf{a}}|\}$ on $\mathbf{V}$, and immediately aborts if $|\psi_{\mathsf{a}}\rangle\langle\psi_{\mathsf{a}}|$ is applied, and otherwise honestly opens (e, r).
3. After completing the protocol, V^* outputs its internal state in $\mathbf{V}$.

It is trivial to construct a simulator for this particular V^* since it just ignores prover's messages. But for explaining our main idea, we examine what happens if we apply the same rewinding strategy as the classical case to the above verifier. After getting a commitment com from V^*, a simulator sends a random a to V^* to extract e. Since we are interested in constructing a simulator that works in the non-aborting case, suppose that V^* does not abort, i.e., sends back a valid opening (e, r). At this point, V^*'s internal state collapses to $|\psi_{\mathsf{na}}\rangle$. Then the simulator cannot "rewind" this state to the original verifier's state $\frac{1}{\sqrt{2}}(|\psi_{\mathsf{a}}\rangle + |\psi_{\mathsf{na}}\rangle)$ in general, and thus the simulation seems to get stuck. However, our key observation is that, conditioned on that V^* does not abort, V^*'s state always collapses to $|\psi_{\mathsf{na}}\rangle$ even in the real execution. Since our goal is to construct $\mathsf{Sim}_{\mathsf{na}}$ that is only required to work for the non-aborting case, it does not matter if V^*'s state collapses to $|\psi_{\mathsf{na}}\rangle$ when the simulator runs extraction. More generally, extraction procedure may collapse verifier's internal state if a similar collapsing happens even in the real execution conditioned on that the verifier does not abort.

Our Idea: Decompose Verifier's Space To generalize the above idea, we want to decompose verifier's internal state after Step 1 into *aborting part* and *non-aborting part*. However, the definition of such a decomposition is non-trivial since a verifier may determine if it aborts depending on the prover's message a in addition to its internal state. Therefore, instead of decomposing it into always-aborting part and always-non-aborting part as in the example of the previous paragraph, we set a noticeable threshold t and decompose it into "not-abort-with-probability $< t$ part" and "not-abort-with-probability $\geq t$ part" over the randomness of a.

For implementing this idea, we rely on Jordan's lemma (e.g., see a lecture note by Regev [AR06]) in a similar way to the work by Nagaj, Wocjan, and Zhang [NWZ09] on the amplification theorem for **QMA**. Let Π be a projection that corresponds to "Step 2 + Step 3 + Check if the verifier does not abort" in GK protocol. A little bit more formally, let $\mathbf{V}$ be a register for verifier's internal state and $\mathbf{Aux}$ be an auxiliary register. Then Π is a projection over $\mathbf{V} \otimes \mathbf{Aux}$ that works as follows:

1. Apply a unitary U_{aux} over $\mathbf{Aux}$ that maps $|0\rangle_{\mathbf{Aux}}$ to $\frac{1}{\sqrt{|\mathcal{R}|}}\sum_{\mathsf{rand}\in\mathcal{R}}|\mathsf{rand}, a_{\mathsf{rand}}\rangle_{\mathbf{Aux}}$ where $\mathcal{R}$ is the randomness space to generate the first message of the Σ-protocol and a_{rand} is the first message derived from the randomness rand.[9]

[8] Strictly speaking, we need to use a slightly stronger variant of collapse-binding commitments which we call *strong collapse-binding* commitments. Such commitments can be constructed under the QLWE assumption or the existence of collapsing hash functions in more general. See Sec. 2.2 for more details.

[9] $\mathbf{Aux}$ stores multiple qubits, but we denote by $|0\rangle_{\mathbf{Aux}}$ to mean $|0^\ell\rangle_{\mathbf{Aux}}$ for the appropriate length ℓ for notational simplicity.

2. Apply a unitary U_V that corresponds to Step 3 for prover's message a_{rand} in $\mathbf{Aux}$ except for measurement,

3. Apply a projection to the subspace spanned by states that contain valid opening (e, r) for com in designated output registers,

4. Apply $(U_V U_{\mathsf{aux}})^\dagger$.

One can see that the probability that the verifier does not abort (i.e., sends a valid opening) is $\|\Pi |\psi\rangle_{\mathbf{V}} |0\rangle_{\mathbf{Aux}}\|^2$ where $|\psi\rangle_{\mathbf{V}}$ is verifier's internal state after Step 1. Then Jordan's lemma gives an orthogonal decomposition of the Hilbert space of $\mathbf{V} \otimes \mathbf{Aux}$ into many one- or two-dimensional subspaces $S_1, ..., S_N$ that are invariant under Π and $|0\rangle_{\mathbf{Aux}} \langle 0|_{\mathbf{Aux}}$ such that we have the following:

1. For any $j \in [N]$ and $|\psi_j\rangle_{\mathbf{V}} |0\rangle_{\mathbf{Aux}} \in S_j$, the projection Π succeeds with probability p_j, i.e., $\|\Pi |\psi_j\rangle_{\mathbf{V}} |0\rangle_{\mathbf{Aux}}\|^2 = p_j$.

2. A success probability of projection Π is "amplifiable" in each subspace. That is, there is an "amplification procedure" Amp that maps any $|\psi_j\rangle_{\mathbf{V}} |0\rangle_{\mathbf{Aux}} \in S_j$ to $\Pi |\psi_j\rangle_{\mathbf{V}} |0\rangle_{\mathbf{Aux}}$ with overwhelming probability within $\mathsf{poly}(\lambda, p_j^{-1})$ times iteration of the same procedure (that does not depend on j) for any $j \in [N]$. Moreover, this procedure does not cause any interference between different subspaces.

Then we define two subspaces

$$S_{<t} := \bigoplus_{j:p_j<t} S_j, \quad S_{\geq t} := \bigoplus_{j:p_j\geq t} S_j.$$

Then for any $|\psi\rangle_{\mathbf{V}}$, we can decompose it as

$$|\psi\rangle_{\mathbf{V}} = |\psi_{<t}\rangle_{\mathbf{V}} + |\psi_{\geq t}\rangle_{\mathbf{V}}$$

by using (sub-normalized) states $|\psi_{<t}\rangle_{\mathbf{V}}$ and $|\psi_{\geq t}\rangle_{\mathbf{V}}$ such that $|\psi_{<t}\rangle_{\mathbf{V}} |0\rangle_{\mathbf{Aux}} \in S_{<t}$ and $|\psi_{\geq t}\rangle_{\mathbf{V}} |0\rangle_{\mathbf{Aux}} \in S_{\geq t}$. In this way, we can formally define a decomposition of verifier's internal state into "not-abort-with-probability $< t$ part" and "not-abort-with-probability $\geq t$ part".

Extraction and Simulation. Then we explain how we can use the above decomposition to implement extraction of e for simulation of non-aborting case. First, we consider an easier case where the verifier's state after Step 1 only has $S_{\geq t}$ component $|\psi_{\geq t}\rangle_{\mathbf{V}}$. In this case, we can use Amp to map $|\psi_{\geq t}\rangle_{\mathbf{V}} |0\rangle_{\mathbf{Aux}}$ onto the span of Π within $\mathsf{poly}(\lambda, t^{-1})$ times iteration. After mapped to Π, we can extract (e, r) without collapsing the state by the definition of Π and our assumption that the commitment is strict-binding. This means that given $|\psi_{\geq t}\rangle_{\mathbf{V}}$, we can extract (e, r), which is information theoretically determined by com, with overwhelming probability. In general, such a deterministic computation can be implemented in a reversible manner, and thus we can extract (e, r) from $|\psi_{\geq t}\rangle_{\mathbf{V}}$ almost without damaging the state.

On the other hand, the same procedure does not work for $|\psi_{<t}\rangle_{\mathbf{V}}$ since $\mathsf{poly}(\lambda, t^{-1})$ times iteration is not sufficient for amplifying the success probability of Π to overwhelming in this subspace. Our idea is to let a simulator run the above extraction procedure in superposition even though $S_{<t}$ component may be damaged.

Specifically, our extraction procedure Ext works as follows:

1. Given a verifier's internal state $|\psi\rangle_{\mathbf{V}}$ after Step 1, initialize $\mathbf{Aux}$ to $|0\rangle_{\mathbf{Aux}}$ and runs Amp for $\mathsf{poly}(\lambda, t^{-1})$ times iteration. Abort if a mapping onto Π does not succeed. Otherwise, proceed to the next step.

2. Apply $U_V U_{\mathsf{aux}}$, measure designated output registers to obtain $(e_{\mathsf{Ext}}, r_{\mathsf{Ext}})$, and apply $(U_V U_{\mathsf{aux}})^\dagger$. We note that $(e_{\mathsf{Ext}}, r_{\mathsf{Ext}})$ is always a valid opening of com since Ext runs this step only if it succeeds in mapping the state onto Π in the previous step. We also note that this step does not collapse the state at all by the strict-binding property of the commitment.

3. Uncompute Step 1 and measure $\mathbf{Aux}$. Abort if the measurement outcome is not 0. Otherwise, proceed to the next step.

4. Output the extracted opening $(e_{\mathsf{Ext}}, r_{\mathsf{Ext}})$ along with a "post-extraction state" $|\psi'\rangle_{\mathbf{V}}$ in register $\mathbf{V}$. For convenience, we express $|\psi'\rangle_{\mathbf{V}}$ as a sub-normalized state whose norm is the probability that Ext does not abort and the post-extraction state conditioned on that the extraction succeeds is $\frac{|\psi'\rangle_{\mathbf{V}}}{\||\psi'\rangle_{\mathbf{V}}\|}$.

In the following, we analyze Ext. We consider the decomposition of $|\psi\rangle_{\mathbf{V}}$ as defined in the previous paragraph:

$$|\psi\rangle_{\mathbf{V}} = |\psi_{<t}\rangle_{\mathbf{V}} + |\psi_{\geq t}\rangle_{\mathbf{V}}\,.$$

Suppose that Ext does not abort, i.e., it outputs a valid opening $(e_{\mathsf{Ext}}, r_{\mathsf{Ext}})$ along with a post-extraction state $|\psi'\rangle_{\mathbf{V}}$. Then, $|\psi'\rangle_{\mathbf{V}}$ can be expressed as

$$|\psi'\rangle_{\mathbf{V}} = |\psi'_{<t}\rangle_{\mathbf{V}} + |\psi'_{\geq t}\rangle_{\mathbf{V}}$$

for some $|\psi'_{<t}\rangle_{\mathbf{V}}$ and $|\psi'_{\geq t}\rangle_{\mathbf{V}}$ such that $|\psi'_{<t}\rangle_{\mathbf{V}}|0\rangle_{\mathbf{Aux}} \in S_{<t}$, $|\psi'_{\geq t}\rangle_{\mathbf{V}}|0\rangle_{\mathbf{Aux}} \in S_{\geq t}$, and $|\psi_{\geq t}\rangle_{\mathbf{V}} \approx |\psi'_{\geq t}\rangle_{\mathbf{V}}$ since there is no interference between $S_{<t}$ and $S_{\geq t}$ when running Amp and $S_{\geq t}$ component hardly changes as observed above. This is not even a close state to the original state $|\psi\rangle_{\mathbf{V}}$ in general since the $S_{<t}$ component may be completely different. However, our key observation is that, conditioned on that the verifier does not abort, at most "t-fraction" of $S_{<t}$ component survives even in the real execution by the definition of the subspace $S_{<t}$. That is, in the verifier's final output state conditioned on that it does not abort, the average squared norm of a portion that comes from $S_{<t}$ component is at most t. Thus, even if a simulator fails to simulate this portion, this only impacts the accuracy of the simulation by a certain function of t, which is shown to be $O(\sqrt{t})$ in the main body.

With this observation in mind, the non-aborting case simulator $\mathsf{Sim}_{\mathsf{na}}$ works as follows.

1. Run Step 1 of the verifier to obtain com and let $|\psi\rangle_{\mathbf{V}}$ be verifier's internal state at this point.

2. Run Ext on input $|\psi\rangle_{\mathbf{V}}$. Abort if Ext aborts. Otherwise, obtain an extracted opening $(e_{\mathsf{Ext}}, r_{\mathsf{Ext}})$ and a post-extraction state $|\psi'\rangle_{\mathbf{V}}$, and proceed to the next step.

3. Simulate a transcript (a, e_{Ext}, z) by the honest-verifier ZK property of the Σ-protocol.

4. Send a to the verifier whose internal state is replaced with $|\psi'\rangle_{\mathbf{V}}$. Let (e, r) be the verifier's response. Abort if (e, r) is not a valid opening to com. Otherwise send z to the verifier.

5. Output the verifier's final output.

By the above analysis, we can see that $\mathsf{Sim}_{\mathsf{na}}$'s output distribution is close to the real verifier's output distribution with an approximation error $O(\sqrt{t})$ conditioned on that the verifier does not abort. Furthermore, the probability that the verifier does not abort can only be changed by at most $O(\sqrt{t})$. If we could set t to be a negligible function, then we would be able to achieve quantum ZK rather than quantum ϵ-ZK. However, since we have to ensure that Amp's running time $\mathsf{poly}(\lambda, t^{-1})$ is polynomial in λ, we can only set t to be noticeable. Since we can set t to be an arbitrarily small noticeable function, we can make the approximation error $O(\sqrt{t})$ be an arbitrarily small noticeable function. This means that the protocol satisfies quantum ϵ-ZK.

Black-Box Simulation. So far, we did not pay attention to the black-box property of simulation. We briefly explain the definition of black-box quantum ZK and that our simulator satisfies it. First, we define black-box quantum ZK by borrowing the definition of quantum oracle machine by Unruh [Unr12]. Roughly, we say that a simulator is black-box if it only accesses unitary part of a verifier and its inverse in a black-box manner, and does not directly act on the verifier's internal registers. With this definition, one part where it is unclear if our simulator is black-box is the amplification procedure Amp. However, by a close inspection, we can see that Amp actually just performs sequential measurements $\{\Pi, I_{\mathbf{V},\mathbf{Aux}} - \Pi\}$ and $\{|0\rangle_{\mathbf{Aux}} \langle 0|_{\mathbf{Aux}}, I_{\mathbf{V},\mathbf{Aux}} - |0\rangle_{\mathbf{Aux}} \langle 0|_{\mathbf{Aux}}\}$, which can be done by black-box access to the verifier as seen from the definition of Π. Therefore, we can see that our simulator is black-box.

A Remark on Underlying Σ-Protocol. In the original GK protocol, any Σ-Protocol can be used as a building block. However, in our technique, we need to use *delayed-witness* Σ-protocol where the first message a can be generated without knowledge of a witness due to a technical reason. An example of delayed-witness Σ-protocol is Blum's Graph Hamiltonicity protocol [Blu86]. Roughly, the reason to require this additional property is for ensuring that a simulator can perfectly simulate the first message a of the Σ-protocol when running the extraction procedure. In the classical setting, a computationally indistinguishable simulation of a works, but we could not prove an analogous claim in our setting.

1.2.2 OWF-based Construction.

Next, we briefly explain our OWF-based quantum ϵ-ZK argument. The reason why we need a stronger assumption in our first construction is that we need to implement the commitment for the challenge by a constant round statistically hiding commitment, which is not known to exist from OWF. Then, a natural idea is to relax it to computationally hiding one if we only need computational soundness. We can show that the extraction technique as explained above also works for statistically binding commitments with a small tweak. However, we cannot prove soundness of the protocol without any modification due to a malleability issue. For explaining this, we recall that the first message a of a Σ-protocol itself is also implemented as a commitment. Then, the computational hiding of commitment does not prevent a computationally bounded prover, which is given a commitment com to e, from generating a "commitment" a whose committed message depends on e. Such a dependence leads to an attack against soundness. To prevent this, an extractable commitment scheme is used to generate a in the classical setting [PW09]. However, since it is unclear if the extractable commitment scheme used in [PW09] is secure against quantum adversaries, we take an alternative approach that we let a prover prove that it knows a committed message inside a by using a proof of knowledge before a verifier opens a challenge as is done in [Gol01, Sec.4.9] (see also [Gol04, App.C.3]). A naive approach to implement this idea would be to use ZK proof of knowledge, but this does not work since a constant round ZK argument is what we are trying to construct. Fortunately, we can instead use witness indistinguishable proof of knowledge (WIPoK) with a simple OR proof trick. Specifically, we let a prover prove that "I know committed message in a" OR "I know witness w for x" where x is the statement being proven in the protocol. In the proof of soundness, since we assume x is a false statement, a witness for the latter statement does not exist. Then we can extract a committed message inside a to break the hiding property of the commitment scheme used by the verifier if the committed message depends on e. On the other hand, in the proof of ϵ-ZK property, we can use the real witness w in an intermediate hybrid to simulate WIPoK without using knowledge of a committed message. In such a hybrid, we can rely on honest-verifier ZK of the Σ-protocol to change a to a simulated one for an extracted

challenge e.

Finally, we remark that though we are not aware of any work that explicitly claims the existence of a constant round WIPoK that works for quantum provers from OWFs, we observe that a combination of known works easily yields such a construction. (See Sec. 2.3.1 for more details.) As a result, we obtain constant round quantum ϵ-ZK argument from OWFs.

1.3 Related Work

ϵ-Zero-Knowledge and Related Notions. Though we are the first to consider ϵ-ZK in the quantum setting, there are several works that consider ϵ-ZK in the classical setting. We briefly review them. We note that all of these results are in the classical setting, and it is unknown if similar results hold in the quantum setting. The notion of ϵ-ZK (originally called ϵ-knowledge) was introduced by Dwork, Naor, and Sahai [DNS04] in the context of concurrent ZK proofs. Bitansky, Kalai, and Paneth [BKP18] gave a construction of 4-round ϵ-ZK proof for **NP** assuming the existence of key-less multi-collision resistant hash function.[10] Barak and Lindell [BL02] showed the impossibility of constant round black-box ZK proof with strict-polynomial time simulation, and observed that strict-polynomial time simulation is possible if we relax ZK to ϵ-ZK. This can be understood as a theoretical separation between ZK and ϵ-ZK. On the other hand, Fleischhacker, Goyal, and Jain [FGJ18] showed that there does not exist 3-round ϵ-ZK proof for **NP** even with non-black-box simulation under some computational assumptions, which is the same lower bound as that for ZK proofs if we allow non-black-box simulation.

Another relaxation of ZK is *super-polynomial simulation (SPS)-ZK* [Pas03], where a simulator is allowed to run in super-polynomial time. One may find a similarity between ϵ-ZK and SPS-ZK in the sense that the latter can be seen as a variant of ϵ-ZK where we set the accuracy parameter ϵ to be negligible. On the other hand, it has been considered that ϵ-ZK is much more difficult to achieve than SPS-ZK. For example, the work of Bitansky, Khurana, and Paneth [BKP19] gave a construction of a 2-round argument for **NP** that achieves a weaker notion of ZK than ϵ-ZK, and the result is considered a significant breakthrough in the area even though there is a simple construction of 2-round SPS-ZK argument for **NP** [Pas03].

Several works considered other weakened notions of ZK [DNRS03, BP12, CLP15, JKKR17, BKP19]. Some of them are weaker than ϵ-ZK, and others are incomparable. For example, "weak ZK" in [BP12, CLP15] is incomparable to ϵ-ZK whereas "weak ZK" in [BKP19] is weaker than ϵ-ZK.

Post-Quantum Zero-Knowledge with Classical Computational Soundness. Ananth and La Placa [AL20] gave a construction of post-quantum ZK argument for **NP** with *classical* computational soundness assuming the QLWE assumption. Though such a protocol would be easy to obtain if we assume average-case classical hardness of certain problems in **BQP** (e.g., factoring) in addition to the QLWE assumption, what is interesting in [AL20] is that they only assume the QLWE assumption.

Post-Quantum Zero-Knowledge with Trusted Setup. Several works studied (non-interactive) post-quantum ZK proofs for **NP** in the common random/reference string model [Kob03, DFS04, PS19]. Among them, Peikert and Shiehian [PS19] proved that there exists non-interactive post-

[10] The protocol achieves full-fledged ZK if we allow the simulator to take non-uniform advice or assume a super-polynomial assumption.

quantum ZK proof for **NP** in the common reference string model assuming the QLWE assumption.[11]

Zero-Knowledge for QMA. The complexity class **QMA** is a quantum analogue of **NP**. Broadbent, Ji, Song, and Watrous [BJSW20] gave a construction of a ZK proof for **QMA**. Recently, Broadbent and Grilo [BG20] gave an alternative simpler construction of a ZK proof for **QMA**. Bitansky and Shmueli [BS20] gave a constant round ZK argument for **QMA** by combining the construction of [BG20] and their post-quantum ZK argument for **NP**. We believe that our technique can be used to construct a constant round ϵ-ZK proof for **QMA** by replacing the delayed-witness Σ-protocol for **NP** with the delayed-witness quantum Σ-protocol for **QMA** recently proposed by Brakerski and Yuen [BY20].[12] This is beyond the scope of this paper, and we leave a formal proof as a future work.

Several works studied non-interactive ZK proofs/arguments for **QMA** in preprocessing models [CVZ20, BG20, Shm20, ACGH20].

Collapsing Hash Functions. The notion of collapsing hash functions was introduced by Unruh [Unr16b] for a replacement of collision-resistant hash functions in post-quantum setting. Unruh [Unr16a] gave a construction of a collapsing hash function under the QLWE assumption. Actually, the construction is generic based on any lossy function with sufficiently large "lossy rate".[13] Currently, we are not aware of any other construction of collapsing hash function based on standard assumptions, but any new construction of collapsing hash function yields a new instantiation of our first construction.

Zhandry [Zha19] proved that any collision-resistant hash function that is not collapsing yields a stronger variant of public-key quantum money (with infinitely often security). Given the difficulty of constructing public key quantum money, he suggested that most natural post-quantum collision-resistant hash functions are likely already collapsing.

Relation to [CCY20]. Our idea of decomposing a verifier's internal space into "aborting space" and "non-aborting space" is inspired by a recent work of Chia, Chung, and Yamakawa [CCY20]. In [CCY20], the authors consider a decomposition of a prover's internal space into "know-answer space" and "not-know-answer space" to prove soundness of parallel repetition version of Mahadev's classical verification of quantum computation protocol [Mah18b]. Though the conceptual idea and some technical tools are similar, the ways of applying them to actual problems are quite different. For example, in our case, we need a careful analysis to make sure that a post-extraction state is close to the original one in some sense while such an argument does not appear in their work since their goal is proving soundness rather than ZK. On the other hand, their technical core is a approximated projection to each subspace, which is not needed in this paper.

Subsequent work. Subsequently to this work, Chia, Chung, Liu, and Yamakawa [CCLY21] proved that there does not exist a constant round post-quantum ZK argument for **NP** unless **NP** $\in$ **BQP**, which is highly unlikely. This justifies the relaxation to ϵ-ZK in our constructions.

[11] In [PS19], they do not explicitly claim ZK against quantum adversaries. However, since their security proof does not rely on rewinding, it immediately extends to post-quantum security if we assume the underlying assumption against quantum adversaries.

[12] Actually, their protocol is delayed-input, i.e., the first message generation does not use the statement either.

[13] A lossy function is defined similarly to a lossy trapdoor function [PW08] except that we do not require the existence of trapdoor.

2 Preliminaries

Basic Notations. We use λ to denote the security parameter throughout the paper. For a positive integer $n \in \mathbb{N}$, $[n]$ denotes a set $\{1, 2, ..., n\}$. For a finite set $\mathcal{X}$, $x \xleftarrow{\$} \mathcal{X}$ means that x is uniformly chosen from $\mathcal{X}$. A function $f : \mathbb{N} \to [0, 1]$ is said to be negligible if for all polynomial p and sufficiently large $\lambda \in \mathbb{N}$, we have $f(\lambda) < 1/p(\lambda)$, said to be overwhelming if $1 - f$ is negligible, and said to be noticeable if there is a polynomial p such that we have $f(\lambda) \geq 1/p(\lambda)$ for sufficiently large $\lambda \in \mathbb{N}$. We denote by poly an unspecified polynomial and by negl an unspecified negligible function. We use PPT and QPT to mean (classical) probabilistic polynomial time and quantum polynomial time, respectively. For a classical probabilistic or quantum algorithm $\mathcal{A}$, $y \xleftarrow{\$} \mathcal{A}(x)$ means that $\mathcal{A}$ is run on input x and outputs y. When $\mathcal{A}$ is classical probabilistic algorithm, we denote by $\mathcal{A}(x; r)$ to mean the execution of $\mathcal{A}$ on input x and a randomness r. When $\mathcal{A}$ is a quantum algorithm that takes a quantum advice, we denote by $\mathcal{A}(x; \rho)$ to mean the execution of $\mathcal{A}$ on input x and an advice ρ. For a quantum algorithm $\mathcal{A}$, a unitary part of $\mathcal{A}$ means the unitary obtained by deferring all measurements by $\mathcal{A}$ and omitting these measurements. We use the bold font (like $\mathbf{X}$) to denote quantum registers, and $\mathcal{H}_{\mathbf{X}}$ to mean the Hilbert space corresponding to the register $\mathbf{X}$. For a quantum state ρ, $M_{\mathbf{X}} \circ \rho$ means a measurement in the computational basis on the register $\mathbf{X}$ of ρ. For quantum states ρ and ρ', $\mathsf{TD}(\rho, \rho')$ denotes trace distance between them. For a pure state $|\psi\rangle$, $\| |\psi\rangle \|$ denotes its Euclidean norm. When we consider a sequence $\{X_\lambda\}_{\lambda \in \mathbb{N}}$ of some objects (e.g., bit strings, quantum states, sets, Hilbert spaces etc.) indexed by the security parameter λ, we often simply write X to mean X_λ or $\{X_\lambda\}_{\lambda \in \mathbb{N}}$, which will be clear from the context. Similarly, for a function f in the security parameter λ, we often simply write f to mean $f(\lambda)$.

Standard Computational Models.

- A PPT algorithm is a probabilistic polynomial time (classical) Turing machine. A PPT algorithm is also often seen as a sequence of uniform polynomial-size circuits.
- A QPT algorithm is a polynomial time quantum Turing machine. A QPT algorithm is also often seen as a sequence of uniform polynomial-size quantum circuits.
- An adversary (or malicious party) is modeled as a non-uniform QPT algorithm $\mathcal{A}$ (with quantum advice) that is specified by sequences of polynomial-size quantum circuits $\{\mathcal{A}_\lambda\}_{\lambda \in \mathbb{N}}$ and polynomial-size quantum advice $\{\rho_\lambda\}_{\lambda \in \mathbb{N}}$. When $\mathcal{A}$ takes an input of λ-bit, $\mathcal{A}$ runs $\mathcal{A}_\lambda$ taking ρ_λ as an advice.

Interactive Quantum Machine and Oracle-Aided Quantum Machine. We rely on the definition of an interactive quantum machine and oracle-aided quantum machine that is given oracle access to an interactive quantum machine following [Unr12]. Roughly, an interactive quantum machine $\mathcal{A}$ is formalized by a unitary over registers $\mathbf{M}$ for receiving and sending messages and $\mathbf{A}$ for maintaining $\mathcal{A}$'s internal state. For two interactive quantum machines $\mathcal{A}$ and $\mathcal{B}$ that share the same message register $\mathbf{M}$, an interaction between $\mathcal{A}$ and $\mathcal{B}$ proceeds by alternating invocations of $\mathcal{A}$ and $\mathcal{B}$ while exchanging messages over $\mathbf{M}$.

An oracle-aided quantum machine $\mathcal{S}$ given oracle access to an interactive quantum machine $\mathcal{A}$ with an initial internal state ρ (denoted by $\mathcal{S}^{\mathcal{A}(\rho)}$) is allowed to apply unitary part of $\mathcal{A}$ and its inverse in a black-box manner where $\mathcal{S}$ can act on $\mathcal{A}$'s internal register $\mathbf{A}$ only through oracle access. We refer to [Unr12] for more formal definitions of interactive quantum machines and black-box access to them.

Indistinguishability of Quantum States. We define computational and statistical indistinguishability of quantum states similarly to [BS20].

We may consider random variables over bit strings or over quantum states. This will be clear from the context. For ensembles of random variables $\mathcal{X} = \{X_i\}_{\lambda\in\mathbb{N},i\in I_\lambda}$ and $\mathcal{Y} = \{Y_i\}_{\lambda\in\mathbb{N},i\in I_\lambda}$ over the same set of indices $I = \bigcup_{\lambda\in\mathbb{N}} I_\lambda$ and a function δ, we write $\mathcal{X} \overset{comp}{\approx}_\delta \mathcal{Y}$ to mean that for any non-uniform QPT algorithm $\mathcal{A} = \{\mathcal{A}_\lambda, \rho_\lambda\}$, there exists a negligible function negl such that for all $\lambda \in \mathbb{N}$, $i \in I_\lambda$, we have

$$|\Pr[\mathcal{A}_\lambda(X_i;\rho_\lambda)] - \Pr[\mathcal{A}_\lambda(Y_i;\rho_\lambda)]| \leq \delta(\lambda) + \mathsf{negl}(\lambda).$$

Especially, when we have the above for $\delta = 0$, we say that $\mathcal{X}$ and $\mathcal{Y}$ are computationally indistinguishable, and simply write $\mathcal{X} \overset{comp}{\approx} \mathcal{Y}$.

Similarly, we write $\mathcal{X} \overset{stat}{\approx}_\delta \mathcal{Y}$ to mean that for any unbounded time algorithm $\mathcal{A}$, there exists a negligible function negl such that for all $\lambda \in \mathbb{N}$, $i \in I_\lambda$, we have

$$|\Pr[\mathcal{A}(X_i)] - \Pr[\mathcal{A}(Y_i)]| \leq \delta(\lambda) + \mathsf{negl}(\lambda).$$[14]

Especially, when we have the above for $\delta = 0$, we say that $\mathcal{X}$ and $\mathcal{Y}$ are statistically indistinguishable, and simply write $\mathcal{X} \overset{stat}{\approx} \mathcal{Y}$. Moreover, we write $\mathcal{X} \equiv \mathcal{Y}$ to mean that X_i and Y_i are distributed identically for all $i \in I$

2.1 Post-Quantum One-Way Functions and Collapsing Hash Functions

A post-quantum one-way function (OWF) is a classically computable function that is hard to invert in QPT. A collapsing hash function is a quantum counterpart of collision-resistant hash function introduced by Unruh [Unr16b]. Unruh [Unr16a] gave a construction of collapsing hash functions based on the QLWE assumption. We give formal definitions in Appendix A since they are only used for constructing other cryptographic primitives and not directly used in our constructions.

2.2 Commitment

We give definitions of commitments and their security. Though mostly standard, we introduce one new security notion which we call *strong collapse-binding*, which is a stronger variant of collapse-biding introduced by Unruh [Unr16b]. As shown in Appendix B, we can see that Unruh's construction of collapse-binding commitments actually also satisfies strong collapse-binding with almost the same (or even simpler) security proof.

Definition 2.1 (Commitment.)**.** *A (two-message) commitment scheme with message space $\mathcal{M}$, randomness space $\mathcal{R}$, commitment space $\mathcal{COM}$, and a public parameter space $\mathcal{PP}$ consists of two classical PPT algorithms* $(\mathsf{Setup}, \mathsf{Commit})$*:*

$\mathsf{Setup}(1^\lambda)$**:** *The setup algorithm takes the security parameter 1^λ as input and outputs a public parameter $\mathsf{pp} \in \mathcal{PP}$.*

$\mathsf{Commit}(\mathsf{pp}, m)$**:** *The committing algorithm takes a public parameter $\mathsf{pp} \in \mathcal{PP}$ and a message $m \in \mathcal{M}$ as input and outputs a commitment $\mathsf{com} \in \mathcal{COM}$.*

[14] In other words, $\mathcal{X} \overset{stat}{\approx}_\delta \mathcal{Y}$ means that there exists a negligible function negl such that the trace distance between ρ_{X_i} and ρ_{Y_i} is at most $\delta(\lambda)+\mathsf{negl}(\lambda)$ for all $\lambda \in \mathbb{N}$ and $i \in I_\lambda$ where ρ_{X_i} and ρ_{Y_i} denote density matrices corresponding to X_i and Y_i.

We say that a commitment scheme is non-interactive if a public parameter pp *generated by* $\mathsf{Setup}(1^\lambda)$ *is always just the security parameter* 1^λ*. For such a scheme, we omit to write* Setup*.*

We define the following security notions for a commitment scheme.

Statistical/Computational Hiding. *For an adversary* $\mathcal{A}$*, we consider an experiment* $\mathsf{Exp}^{\mathsf{hiding}}_{\mathcal{A}}(1^\lambda)$ *defined below:*

1. $\mathcal{A}$ *is given the security parameter* 1^λ *and sends a (possibly malformed) public parameter* $\mathsf{pp} \in \mathcal{PP}$ *and* $(m_0, m_1) \in \mathcal{M}^2$ *to the challenger*
2. *The challenger randomly picks* $b \xleftarrow{\$} \{0,1\}$*, computes* $\mathsf{com} \xleftarrow{\$} \mathsf{Commit}(\mathsf{pp}, m_b)$*, and sends* com *to* $\mathcal{A}$*.*
3. $\mathcal{A}$ *is given a commitment* com *and outputs* $b' \in \{0,1\}$*. The experiment outputs* 1 *if* $b = b'$ *and* 0 *otherwise.*

We say that a commitment scheme satisfies statistical (resp. computational) hiding if for any unbounded-time (resp. non-uniform QPT) adversary $\mathcal{A}$*, we have*

$$|\Pr[1 \xleftarrow{\$} \mathsf{Exp}^{\mathsf{hiding}}_{\mathcal{A}}(1^\lambda)] - 1/2| = \mathsf{negl}(\lambda).$$

Remark 1. *Our definition of hiding requires that the security should hold even if* pp *is maliciously generated. Thus, the hiding property holds even if a receiver runs the setup algorithm.*

Binding.

- ***Perfect/Statistical/Computational Binding.*** *We say that a non-interactive commitment scheme satisfies statistical (resp. computational) binding if for any unbounded-time (resp. non-uniform QPT) adversary* $\mathcal{A}$*, we have*

  $$\Pr[\mathsf{Commit}(\mathsf{pp}, m; r) = \mathsf{Commit}(\mathsf{pp}, m'; r') \wedge m \neq m' : \mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(1^\lambda), (m, m', r, r') \xleftarrow{\$} \mathcal{A}(\mathsf{pp})] = \mathsf{negl}(\lambda).$$

  We say that a scheme satisfies perfect binding if the above probability is 0 *for all unbounded-time adversary* $\mathcal{A}$*.*

- ***Strong Collapse-Binding.*** *For an adversary* $\mathcal{A}$*, we define an experiment* $\mathsf{Exp}^{\mathsf{cl\text{-}binding}}_{\mathcal{A}}(1^\lambda)$ *as follows:*

  1. *The challenger generates* $\mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$*.*
  2. $\mathcal{A}$ *is given the public parameter* pp *as input and generates a commitment* $\mathsf{com} \in \mathcal{COM}$ *and a quantum state* σ *over registers* $(\mathbf{M}, \mathbf{R}, \mathbf{A})$ *where* $\mathbf{M}$ *stores an element of* $\mathcal{M}$*,* $\mathbf{R}$ *stores an element of* $\mathcal{R}$*, and* $\mathbf{A}$ *is* $\mathcal{A}$*'s internal register. Then it sends* com *and registers* $(\mathbf{M}, \mathbf{R})$ *to the challenger, and keeps* $\mathbf{A}$ *on its side.*
  3. *The challenger picks* $b \xleftarrow{\$} \{0,1\}$ *If* $b = 0$*, the challenger does nothing and if* $b = 1$*, the challenger measures registers* $(\mathbf{M}, \mathbf{R})$ *in the computational basis. The challenger returns registers* $(\mathbf{M}, \mathbf{R})$ *to* $\mathcal{A}$
  4. $\mathcal{A}$ *outputs a bit* b'*. The experiment outputs* 1 *if* $b' = b$ *and* 0 *otherwise.*

We say that $\mathcal{A}$ is a valid adversary if we we have

$$\Pr[\mathsf{Commit}(\mathsf{pp}, m; r) = \mathsf{com} : \mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(1^\lambda), (\mathsf{com}, \sigma) \xleftarrow{\$} \mathcal{A}(\mathsf{pp}), (m, r) \leftarrow M_{\mathbf{M},\mathbf{R}} \circ \sigma] = 1.$$

We say that a commitment is strongly collapse-binding if for any non-uniform QPT valid adversary $\mathcal{A}$, we have

$$|\Pr[1 \xleftarrow{\$} \mathsf{Exp}^{\mathsf{cl\text{-}binding}}_{\mathcal{A}}(1^\lambda)] - 1/2| = \mathsf{negl}(\lambda).$$

Remark 2. *The difference of strong collapse-binding from the original collapse-binding is that the challenger measures both registers $(\mathbf{M}, \mathbf{R})$ in Step 3 in the case of $b = 1$ whereas the challenger of the original collapse-binding game only measures $\mathbf{M}$. We note that the statistical binding property immediately implies the (original) collapse-binding property, but it does not imply the strong collapse-binding property.*

Remark 3. *One can easily see that the strong collapse-binding property implies the computational binding property. Indeed, if one can find $(m, r) \neq (m', r')$ such that $\mathsf{Commit}(\mathsf{pp}, m; r) = \mathsf{Commit}(\mathsf{pp}, m'; r') = \mathsf{com}$, then we can break the strong collapse-binding property by sending com and $|\psi\rangle := \frac{1}{\sqrt{2}}(|m, r\rangle_{\mathbf{M},\mathbf{R}} + |m', r'\rangle_{\mathbf{M},\mathbf{R}})$ to the challenger and performing a measurement $(|\psi\rangle\langle\psi|, I - |\psi\rangle\langle\psi|)$ on the returned state to distinguish if the state is measured.*

We introduce the following definition for convenience.

Definition 2.2 (Binding Public Parameter.)**.** *We say that $\mathsf{pp} \in \mathcal{PP}$ is* binding *if for any commitment $\mathsf{com} \in \mathcal{COM}$, there is at most one $m \in \mathcal{M}$ such that $\mathsf{Commit}(\mathsf{pp}, m; r) = \mathsf{com}$ for some $r \in \mathcal{R}$.*

The following lemma is easy to see.

Lemma 2.3. *If a commitment scheme is statistically binding, then overwhelming fraction of pp generated by $\mathsf{Setup}(1^\lambda)$ is binding.*

We also consider an additional security definition.

Definition 2.4 (Unpredictability.)**.** *For an adversary $\mathcal{A}$, we consider an experiment $\mathsf{Exp}^{\mathsf{unpre}}_{\mathcal{A}}(1^\lambda)$ defined below:*

1. *$\mathcal{A}$ is given the security parameter 1^λ and sends a (possibly malformed) public parameter $\mathsf{pp} \in \mathcal{PP}$ to the challenger.*
2. *The challenger randomly picks $m \xleftarrow{\$} \mathcal{M}$, computes $\mathsf{com} \xleftarrow{\$} \mathsf{Commit}(\mathsf{pp}, m)$, and sends com to $\mathcal{A}$.*
3. *$\mathcal{A}$ returns m^*. The experiment outputs 1 if $m = m^*$ and 0 otherwise.*

We say that a commitment scheme is unpredictable if for any non-uniform QPT adversary $\mathcal{A}$, we have

$$\Pr[1 \xleftarrow{\$} \mathsf{Exp}^{\mathsf{unpre}}_{\mathcal{A}}(1^\lambda)] = \mathsf{negl}(\lambda).$$

The following lemma is a folklore, and easy to prove.

Lemma 2.5. *If a commitment scheme is computationally hiding and $|\mathcal{M}| = 2^{\omega(\lambda)}$, then the scheme is unpredictable.*

Instantiations. A computationally hiding and statistically binding commitment scheme exists under the existence of OWF [Nao91, HILL99]. A computationally hiding and perfectly binding non-interactive commitment scheme exists under the QLWE assumption [GHKW17, LS19].

A statistically hiding and strong collapse-binding commitment scheme exists assuming the existence of collapsing hash functions (and thus under the QLWE assumption) [Unr16b, Unr16a]. This can be seen by observing that the proof of (original) collapse-binding property from collapsing hash functions in [Unr16b, Unr16a] already implicitly proves the strong collapse-binding property. For completeness, we give a proof in Appendix B.

2.3 Interactive Proof and Argument.

We define interactive proofs and arguments similarly to [BS20].

Notations. For an **NP** language L and $x \in L$, $R_L(x)$ is the set that consists of all (classical) witnesses w such that the verification machine for L accepts (x, w).

A (classical) interactive protocol is modeled as an interaction between interactive quantum machines P referred to as a prover and V referred to as a verifier that can be implemented by PPT algorithms. We denote by $\langle P(x_P), V(x_V)\rangle(x)$ an execution of the protocol where x is a common input, x_P is P's private input, and x_V is V's private input. We denote by $\mathsf{OUT}_V\langle P(x_P), V(x_V)\rangle(x)$ the fianl output of V in the execution. An honest verifier's output is $\top$ indicating acceptance or $\bot$ indicating rejection, and a quantum malicious verifier's output may be an arbitrary quantum state.

Definition 2.6 (Interactive Proof and Argument for **NP**)**.** *An interactive proof or argument for an* **NP** *language L is an interactive protocol between a PPT prover P and a PPT verifier V that satisfies the following:*

Perfect Completeness. *For any $x \in L$, and $w \in R_L(x)$, we have*

$$\Pr[\mathsf{OUT}_V\langle P(w), V\rangle(x) = \top] = 1$$

Statistical/Computational Soundness. *We say that an interactive protocol is statistically (resp. computationally) sound if for any unbounded-time (resp. non-uniform QPT) cheating prover P^*, there exists a negligible function negl such that for any $\lambda \in \mathbb{N}$ and any $x \in \{0,1\}^\lambda \setminus L$, we have*

$$\Pr[\mathsf{OUT}_V\langle P^*, V\rangle(x) = \top] \leq \mathsf{negl}(\lambda).$$

We call an interactive protocol with statistical (resp. computational) soundness an interactive proof (resp. argument).

2.3.1 Witness Indistinguishable Proof of Knowledge

Definition 2.7 (Witness Indistinguishable Proof of Knowledge)**.** *A witness indistinguishable proof of knowledge for an* **NP** *language L is an interactive proof for L that satisfies the following properties (in addition to perfect completeness and statistical soundness):*

Witness Indistinguishability. *For any non-uniform QPT malicious verifier V^*, we have*

$$\{\mathsf{OUT}_{V^*}\langle P(w_0), V^*\rangle(x)\}_{\lambda,x,w_0,w_1} \overset{comp}{\approx} \{\mathsf{OUT}_{V^*}\langle P(w_1), V^*\rangle(x)\}_{\lambda,x,w_0,w_1}$$

where $\lambda \in \mathbb{N}$, $x \in L \cap \{0,1\}^\lambda$, and $w_0, w_1 \in R_L(x)$.

(Non-Adaptive) Knowledge Extractability. *There is an oracle-aided QPT algorithm $\mathcal{K}$, a polynomial* poly*, a negligible function* negl*, and a constant $d \in \mathbb{N}$ such that for any quantum unbounded-time malicious prover $P^* = \{P^*_\lambda, \rho_\lambda\}_{\lambda\in\mathbb{N}}$, $\lambda \in \mathbb{N}$, and $x \in \{0,1\}^\lambda$, we have*

$$\Pr[w \in R_L(x) : w \xleftarrow{\$} \mathcal{K}^{P^*_\lambda(\rho_\lambda)}(x)] \geq \frac{1}{\mathsf{poly}(\lambda)} \cdot \Pr[\mathsf{OUT}_V\langle P^*_\lambda(\rho_\lambda), V\rangle(x) = \top]^d - \mathsf{negl}(\lambda).$$

Instantiations. We can construct a constant round (actually 4-round) witness indistinguishable proof of knowledge for **NP** only assuming the existence of post-quantum OWFs by some tweak of existing works [Unr12, Unr16b]. We briefly explain this below.

A constant round witness indistinguishable proof of knowledge that satisfies the above requirements was first constructed by Unruh [Unr12] based on *strict-binding commitments*, where a commitment perfectly binds not only a message but also randomness. Due to the usage of strict-binding commitment, an instantiation of this protocol requires one-to-one OWF, for which there is no post-quantum candidate under standard assumptions. Later, Unruh [Unr16b] proved that the protocol in [Unr12] can be instantiated using collapse-binding commitments instead of strict-binding commitments if we relax the knowledge extractability requirement to computational one. Since the statistical binding property trivially implies the collapse-binding property as noted in Remark 2, we can just use statistically binding commitments as collapse-binding commitments in the construction of [Unr16b]. Moreover, since the statistically binding property can be seen as a "statistical version" of collapse-binding, we obtain statistical knowledge extractability.[15] In summary, the construction in [Unr16b] instantiated with statistically binding (and computational hiding) commitments suffices for our purpose.

We also give another more concrete explanation. The protocol in [Unr12] is a modification of Blum's Graph Hamiltonicity protocol [Blu86]. For proving the knowledge extractability, Unruh introduced a rewinding technique that enables the extractor to run a prover twice for different challenges. In his extraction strategy, the extractor records *both committed messages and randomness* when it runs the prover for the first time. For ensuring that this does not collapse the prover's state too much, he assumed the strict-binding property. Here, we observe that the extractor actually need not record both committed message and randomness, and it only need to record the committed message. (Indeed, the security proof in [Unr16b] does so). In this case, the statistical binding property instead of the strict-binding property suffices to ensure that the prover's state is not collapsed too much since the randomness register is not measured by the extractor.

2.3.2 Delayed-Witness Σ-Protocol

We introduce a special type of Σ-protocol which we call *delayed-witness Σ-protocol* where the first message can be generated without witness.

Definition 2.8 (Delayed-Witness Σ-protocol)**.** *A (post-quantum) delayed-witness Σ-protocol for an* **NP** *language L is a 3-round interactive proof for* **NP** *with the following syntax.*

Common Input: An instance $x \in L \cap \{0,1\}^\lambda$ for security parameter $\lambda \in \mathbb{N}$.
P's Private Input: A classical witness $w \in R_L(x)$ for x.

1. P generates a "commitment" a and a state st. For this part, P only uses the statement x and does not use any witness w. We denote this procedure by $(a, \mathsf{st}) \xleftarrow{\$} \Sigma.P_1(x)$. Then it sends a to the verifier, and keeps st as its internal state.

[15] If one is not convinced by this informal explanation, one can think of our claim as the existence of a constant round witness indistinguishable *argument* of knowledge for **NP** under the existence of OWF. Actually, this computational version of knowledge extractability suffices for the purpose of this paper.

2. V chooses a"challenge" $e \xleftarrow{\$} \{0,1\}^\lambda$ and sends e to P.

3. P generates a "response" z from st, witness w, and e. We denote this procedure by $z \xleftarrow{\$} \Sigma.P_3(\mathsf{st}, w, e)$. Then it sends z to V.

4. V verifies the transcript (a, e, z) and outputs $\top$ indicating acceptance or $\bot$ indicating rejection. We denote this procedure by $\top/\bot \xleftarrow{\$} \Sigma.V(x, a, e, z)$.

We require a delayed-witness Σ-protocol to satisfy the following property in addition to perfect completeness and statistical soundness.[16]

Special Honest-Verifier Zero-Knowledge. There exists a PPT simulator Sim_Σ such that we have

$$\{(a,z) : (a,\mathsf{st}) \xleftarrow{\$} \Sigma.P_1(x), z \xleftarrow{\$} \Sigma.P_3(\mathsf{st}, w, e)\}_{\lambda,x,w,e} \overset{comp}{\approx} \{(a,z) : (a,z) \xleftarrow{\$} \mathsf{Sim}_\Sigma(x,e)\}_{\lambda,x,w,e}$$

where $x \in L \cap \{0,1\}^\lambda$, $w \in R_L(x)$, and $e \in \{0,1\}^\lambda$.

Instantiations. An example of a dealyed-witness Σ-protocol is a parallel repetition version of Blum's Graph Hamiltonicity protocol [Blu86]. In the ptorocol, we need a computationally hiding and perfectly binding non-interactive commitment scheme, which exists under the QLWE assumption as noted in Sec. 2.2. In summary, a delayed-input Σ-protocol for all **NP** languages exists under the QLWE assumption.

2.3.3 Quantum ϵ-Zero-Knowledge Proof and Argument

Here, we define quantum black-box ϵ-zero-knowledge proofs and arguments. The difference from the definition of quantum zero-knowledge in [BS20] are:

1. (ϵ-**Zero-Knowledge**) We allow the simulator to depend on a noticeable "accuracy parameter" ϵ, and allows its running time to polynomially depend on ϵ^{-1}, and

2. (**Black-Box Simulation**) the simulator is only given black-box access to a malicious verifier.

Definition 2.9 (Post-Quantum Black-Box ϵ-Zero-Knowledge Proof and Argument)**.** *A post-quantum black-box ϵ-zero-knowledge proof (resp. argument) for an* **NP** *language L is an interactive proof (resp. argument) for L that satisfies the following property in addition to perfect completeness and statistical (resp. computational) soundness:*

Quantum Black-Box ϵ-Zero-Knowledge. *There exists an oracle-aided QPT simulator* Sim *such that for any non-uniform QPT malicious verifier $V^* = \{V^*_\lambda, \rho_\lambda\}_{\lambda\in\mathbb{N}}$ and any noticeable function $\epsilon(\lambda)$, we have*

$$\{\mathsf{OUT}_{V^*_\lambda}\langle P(w), V^*_\lambda(\rho_\lambda)\rangle(x)\}_{\lambda,x,w} \overset{comp}{\approx}_\epsilon \{\mathsf{OUT}_{V^*_\lambda}(\mathsf{Sim}^{V^*_\lambda(\rho_\lambda)}(x, 1^{\epsilon^{-1}}))\}_{\lambda,x,w}$$

where $\lambda \in \mathbb{N}$, $x \in L \cap \{0,1\}^\lambda$, $w \in R_L(\lambda)$, and $\mathsf{OUT}_{V^*_\lambda}(\mathsf{Sim}^{V^*_\lambda(\rho_\lambda)}(x))$ is the state in the output register of V^*_λ after the simulated execution of V^*_λ by Sim.

[16] We do not require *special soundness*, which is often a default requirement of Σ-protocol.

Remark 4. *In the above definition of quantum black-box ϵ-zero-knowledge, we do not consider an entanglement between auxiliary input of a malicious verifier and distinguisher unlike the original definition of quantum zero-knowledge by Watrous [Wat09]. However, in Appendix C, we show that the above definition implies indistinguishability against a distinguisher that may get an entangled state to verifier's auxiliary input by taking advantage of black-box simulation.*

2.4 Quantum Rewinding Lemma

Watrous [Wat09] proved a lemma that enables us to amplify the success probability of a quantum algorithm under certain conditions. The following form of the lemma is based on that in [BS20, Lemma 2.1].

Lemma 2.10 ([Wat09, BS20])**.** *There is an oracle-aided quantum algorithm R that gets as input the following:*

- *A quantum circuit Q that takes n-input qubits in register Inp and outputs a classical bit b (in a register outside Inp) and an m output qubits.*
- *An n-qubit state ρ in register Inp.*
- *A number $T \in \mathbb{N}$ in unary.*

$\mathsf{R}(1^T, \mathsf{Q}, \rho)$ executes in time $T \cdot |\mathsf{Q}|$ and outputs a distribution over m-qubit states $D_\rho := \mathsf{R}(1^T, \mathsf{Q}, \rho)$ with the following guarantees.

For an n-qubit state ρ, denote by Q_ρ the conditional distribution of the output distribution $\mathsf{Q}(\rho)$, conditioned on $b = 0$, and denote by $p(\rho)$ the probability that $b = 0$. If there exist $p_0, q \in (0,1)$, $\gamma \in (0, \frac{1}{2})$ such that:

- *Amplification executes for enough time: $T \geq \frac{\log(1/\gamma)}{4p_0(1-p_0)}$,*
- *There is some minimal probability that $b = 0$: For every n-qubit state ρ, $p_0 \leq p(\rho)$,*
- *$p(\rho)$ is input-independent, up to γ distance: For every n-qubit state ρ, $|p(\rho) - q| < \gamma$, and*
- *q is closer to $\frac{1}{2}$: $p_0(1-p_0) \leq q(1-q)$,*

then for every n-qubit state ρ,

$$\mathsf{TD}(\mathsf{Q}_\rho, D_\rho) \leq 4\sqrt{\gamma}\frac{\log(1/\gamma)}{p_0(1-p_0)}.$$

Moreover, $\mathsf{R}(1^T, \mathsf{Q}, \rho)$ works in the following manner: It uses Q for only implementing oracles that perform the unitary part of Q and its inverse, acts on Inp only through these oracles, and the output of R is the state in the output register of Q after the simulated execution. We note that R may directly act on Q's internal registers other than Inp.

Remark 5. *The final claim of the lemma ("Moreover...") is not explicitly stated in previous works. In the description of R in [Wat09], the first qubit of Inp is designated to output b, and thus the above requirement is not satisfied. However, this can be easily avoided by just letting Q output b in a register outside Inp as required above. Then one can see that R acts on the input register only through Q as seen from the description of R in [Wat09] (with the above modification in mind). Looking ahead, this is needed to show our ϵ-zero-knowledge simulators are black-box.*

3 Technical Lemmas

In this section, we introduce three lemmas that are used in the proof of the extraction lemma (Lemma 4.2) in Sec. 4.

Lemma 3.1. *Let $|\phi_b\rangle = |\phi_{b,0}\rangle + |\phi_{b,1}\rangle$ be a normalized quantum state and Π be a projector over a Hilbert space $\mathcal{H}$ such that $\langle\phi_{b,0}|\Pi|\phi_{b',1}\rangle = 0$ for $b, b' \in \{0,1\}$, $\||\Pi|\phi_{b,0}\rangle\|^2 \leq \gamma$ for $b \in \{0,1\}$, and $\| |\phi_{1,1}\rangle - |\phi_{0,1}\rangle \| \leq \delta$ for some real numbers γ, δ. Let F be a quantum algorithm that takes a state in $\mathcal{H}$ as input, applies the projective measurement $(\Pi, I-\Pi)$, and outputs the resulting state if the measurement outcome is 0 i.e., the state is projected onto Π, and otherwise outputs $\perp$.*

Then it holds that

$$\mathsf{TD}(F(|\phi_0\rangle), F(|\phi_1\rangle)) \leq \sqrt{4\gamma + 2\delta}.$$

Proof. If $\sqrt{4\gamma+2\delta} > 1$, then the desired inequality trivially holds. Thus, we assume $\sqrt{4\gamma+2\delta} \leq 1$ in the rest of the proof. We consider an additional one-qubit register and define

$$|\psi_b\rangle := \sqrt{1-p_b}\,|0\rangle\,|0^m\rangle + |1\rangle\,\Pi\,|\phi_b\rangle$$

for $b \in \{0,1\}$ where m is the number of qubits in the register for $|\phi_b\rangle$ and

$$p_b := \||\Pi|\phi_b\rangle\|^2.$$

Without loss of generality, we assume $p_0 \geq p_1$. It suffices to prove

$$\mathsf{TD}(|\psi_0\rangle\langle\psi_0|, |\psi_1\rangle\langle\psi_1|) \leq \sqrt{4\gamma+2\delta} \tag{1}$$

because a distinguisher that distinguishes $F(|\phi_0\rangle)$ and $F(|\phi_1\rangle)$ can be easily converted into a distinguisher that distinguishes $|\psi_0\rangle$ and $|\psi_1\rangle$ with the same advantage.

We have

$$\begin{aligned}
p_0 &= \||\Pi|\phi_0\rangle\|^2 \\
&= \||\Pi|\phi_{0,0}\rangle\|^2 + \||\Pi|\phi_{0,1}\rangle\|^2 \\
&\leq \||\Pi|\phi_{0,1}\rangle\|^2 + \gamma
\end{aligned}$$

where we used the assumption that $\langle\phi_{0,0}|\Pi|\phi_{0,1}\rangle = 0$ in the second equality and the assumption that $\||\Pi|\phi_{0,0}\rangle\|^2 \leq \gamma$ in the final inequality.

Thus, we have

$$\||\Pi|\phi_{0,1}\rangle\|^2 \geq p_0 - \gamma. \tag{2}$$

We give a lower bound for $|\langle\psi_0|\psi_1\rangle|$. By the definition of $|\psi_b\rangle$,

$$\begin{aligned}
|\langle\psi_0|\psi_1\rangle| &= |\sqrt{(1-p_0)(1-p_1)} + \langle\phi_0|\Pi|\phi_1\rangle| \\
&= |\sqrt{(1-p_0)(1-p_1)} + \langle\phi_{0,0}|\Pi|\phi_{1,0}\rangle + \langle\phi_{0,1}|\Pi|\phi_{1,1}\rangle| \\
&= |\sqrt{(1-p_0)(1-p_1)} + \langle\phi_{0,0}|\Pi|\phi_{1,0}\rangle + \langle\phi_{0,1}|\Pi|\phi_{0,1}\rangle + \langle\phi_{0,1}|\Pi(|\phi_{1,1}\rangle - |\phi_{0,1}\rangle)| \\
&\geq (1-p_0) + \||\Pi|\phi_{0,1}\rangle\|^2 - \||\Pi|\phi_{0,0}\rangle\| \cdot \||\Pi|\phi_{1,0}\rangle\| - \| |\phi_{1,1}\rangle - |\phi_{0,1}\rangle \| \\
&\geq (1-p_0) + (p_0 - \gamma) - \gamma - \delta \\
&= 1 - (2\gamma + \delta)
\end{aligned}$$

where we used the assumption that $\langle\phi_{b,0}|\,\Pi\,|\phi_{b',1}\rangle = 0$ for $b, b' \in \{0,1\}$ in the first equality, the assumptions that $p_0 \geq p_1$ and $|\phi_{0,1}\rangle = |\phi_{1,1}\rangle$ in the first inequality, and Eq. 2 and the assumptions that $\|\Pi\,|\phi_{b,0}\rangle\|^2 \leq \gamma$ for $b \in \{0,1\}$ and $\|\,|\phi_{1,1}\rangle - |\phi_{0,1}\rangle\,\| \leq \delta$ in the second inequality. We note that $1-(2\gamma+\delta) > 0$ since we assume $\sqrt{4\gamma+2\delta} \leq 1$.

Then, we have

$$\begin{aligned}\mathsf{TD}(|\psi_0\rangle\langle\psi_0|, |\psi_1\rangle\langle\psi_1|) &= \sqrt{1-|\langle\psi_0|\psi_1\rangle|^2}\\ &\leq \sqrt{4\gamma+2\delta}\end{aligned}$$

This completes the proof of Lemma 3.1.

□

The second lemma is the following variant of the gentle measurement lemma.

Lemma 3.2. *Let $|\psi\rangle_{\mathbf{X}}$ be a (not necessarily normalized) state over register $\mathbf{X}$ and U be a unitary over registers $(\mathbf{X},\mathbf{Y},\mathbf{Z})$. Suppose that a measurement of register $\mathbf{Z}$ of $U\,|\psi\rangle_{\mathbf{X}}\,|0\rangle_{\mathbf{Y},\mathbf{Z}}$ results in a deterministic value except for probability ν, i.e., there is z^* such that*

$$\|(I - |z^*\rangle\langle z^*|_{\mathbf{Z}})U\,|\psi\rangle_{\mathbf{X}}\,|0\rangle_{\mathbf{Y},\mathbf{Z}}\,\|^2 \leq \nu.$$

If we let $R := (|0\rangle\langle 0|)_{\mathbf{Y},\mathbf{Z}}U^\dagger(|z^*\rangle\langle z^*|)_{\mathbf{Z}}U$, then we have

$$\|\,|\psi\rangle_{\mathbf{X}}\,|0\rangle_{\mathbf{Y},\mathbf{Z}} - R\,|\psi\rangle_{\mathbf{X}}\,|0\rangle_{\mathbf{Y},\mathbf{Z}}\,\| \leq \sqrt{\nu}.$$

Proof. Let $\Pi_{z^*} := (|z^*\rangle\langle z^*|)_{\mathbf{Z}}$. We have

$$\begin{aligned}|\psi\rangle_{\mathbf{X}}\,|0\rangle_{\mathbf{Y},\mathbf{Z}} &= (|0\rangle\langle 0|)_{\mathbf{Y},\mathbf{Z}}U^\dagger U\,|\psi\rangle_{\mathbf{X}}\,|0\rangle_{\mathbf{Y},\mathbf{Z}}\\ &= R\,|\psi\rangle_{\mathbf{X}}\,|0\rangle_{\mathbf{Y},\mathbf{Z}} + (|0\rangle\langle 0|)_{\mathbf{Y},\mathbf{Z}}U^\dagger(I-\Pi_{z^*})U\,|\psi\rangle_{\mathbf{X}}\,|0\rangle_{\mathbf{Y},\mathbf{Z}}\,.\end{aligned}$$

Thus, we have

$$\begin{aligned}\|\,|\psi\rangle_{\mathbf{X}}\,|0\rangle_{\mathbf{Y},\mathbf{Z}} - R\,|\psi\rangle_{\mathbf{X}}\,|0\rangle_{\mathbf{Y},\mathbf{Z}}\,\| &= \|(|0\rangle\langle 0|)_{\mathbf{Y},\mathbf{Z}}U^\dagger(I-\Pi_{z^*})U\,|\psi\rangle_{\mathbf{X}}\,|0\rangle_{\mathbf{Y},\mathbf{Z}}\,\|\\ &\leq \|(I-\Pi_{z^*})U\,|\psi\rangle_{\mathbf{X}}\,|0\rangle_{\mathbf{Y},\mathbf{Z}}\,\| \leq \sqrt{\nu}.\end{aligned}$$

□

The third lemma is about amplifying the success probability of a projection. Very roughly speaking, the lemma states that for any projection Π and a "threshold" $0 < t < 1$, we can decompose the Hilbert space into two subspaces $S_{<t}$ and $S_{\geq t}$ so that

1. Π succeeds with probability $< t$ (resp. $\geq t$) in $S_{<t}$ (resp. $S_{\geq t}$).
2. There is an efficient procedure Amp that runs in time $T = O(t^{-1})$ that maps any state $|\psi\rangle \in S_{\geq t}$ onto the span of Π with probability almost 1. We note that this does not necessarily map $|\psi\rangle$ to $\Pi\,|\psi\rangle$.
3. Each subspace is invariant under Π and Amp.

The formal statement of our lemma is given below:

Lemma 3.3. *Let Π be a projection over a Hilbert space $\mathcal{H}_{\mathbf{X}} \otimes \mathcal{H}_{\mathbf{Y}}$. For any noticeable function $t = t(\lambda)$, there exists an orthogonal decomposition $(S_{<t}, S_{\geq t})$ of $\mathcal{H}_{\mathbf{X}} \otimes \mathcal{H}_{\mathbf{Y}}$ that satisfies the following:*

1. *($S_{<t}$ **and** $S_{\geq t}$ **are invariant under** Π **and** $(|0\rangle\langle 0|)_{\mathbf{Y}}$.) For any $|\psi\rangle_{\mathbf{X},\mathbf{Y}} \in S_{<t}$, we have*

$$\Pi|\psi\rangle_{\mathbf{X},\mathbf{Y}} \in S_{<t}, \qquad (I_{\mathbf{X}} \otimes (|0\rangle\langle 0|)_{\mathbf{Y}})|\psi\rangle_{\mathbf{X},\mathbf{Y}} \in S_{<t}.$$

Similarly, for any $|\psi\rangle_{\mathbf{X},\mathbf{Y}} \in S_{\geq t}$, we have

$$\Pi|\psi\rangle_{\mathbf{X},\mathbf{Y}} \in S_{\geq t}, \qquad (I_{\mathbf{X}} \otimes (|0\rangle\langle 0|)_{\mathbf{Y}})|\psi\rangle_{\mathbf{X},\mathbf{Y}} \in S_{\geq t}.$$

2. *(Π **succeeds with probability** $< t$ **and** $\geq t$ **in** $S_{<t}$ **and** $S_{\geq t}$.) For any quantum state $|\phi\rangle_{\mathbf{X}} \in \mathcal{H}_{\mathbf{X}}$ s.t. $|\phi\rangle_{\mathbf{X}}|0\rangle_{\mathbf{Y}} \in S_{<t}$ we have*

$$\|\Pi|\phi\rangle_{\mathbf{X}}|0\rangle_{\mathbf{Y}}\|^2 < t.$$

Similarly, for any quantum state $|\phi\rangle_{\mathbf{X}} \in \mathcal{H}_{\mathbf{X}}$ s.t. $|\phi\rangle_{\mathbf{X}}|0\rangle_{\mathbf{Y}} \in S_{\geq t}$ we have

$$\|\Pi|\phi\rangle_{\mathbf{X}}|0\rangle_{\mathbf{Y}}\|^2 \geq t.$$

3. *(**Unitary for amplification.**) For any $T \in \mathbb{N}$, there exists a unitary $U_{\mathsf{amp},T}$ over $\mathcal{H}_{\mathbf{X}} \otimes \mathcal{H}_{\mathbf{Y}} \otimes \mathcal{H}_{\mathbf{B}} \otimes \mathcal{H}_{\mathbf{Anc}}$ where $\mathbf{B}$ is a register to store a qubit and $\mathbf{Anc}$ is a register to store ancillary qubits with the following properties:*

   (a) *(**Mapped onto** Π **when** $\mathbf{B}$ **contains** 1.) For any quantum state $|\psi\rangle_{\mathbf{X},\mathbf{Y}} \in \mathcal{H}_{\mathbf{X}} \otimes \mathcal{H}_{\mathbf{Y}}$, we can write*

$$|1\rangle\langle 1|_{\mathbf{B}} U_{\mathsf{amp},T}|\psi\rangle_{\mathbf{X},\mathbf{Y}}|0\rangle_{\mathbf{B},\mathbf{Anc}} = \sum_{anc} |\psi'_{anc}\rangle_{\mathbf{X},\mathbf{Y}}|1\rangle_{\mathbf{B}}|anc\rangle_{\mathbf{Anc}}$$

   by using sub-normalized states $|\psi'_{anc}\rangle_{\mathbf{X},\mathbf{Y}}$ that are in the span of Π.

   (b) *(**Amplification of success probability in** $S_{\geq t}$.) For any noticeable function $\nu = \nu(\lambda)$, there is $T = \mathsf{poly}(\lambda)$ such that for any quantum state $|\phi\rangle_{\mathbf{X}} \in \mathcal{H}_{\mathbf{X}}$ s.t. $|\phi\rangle_{\mathbf{X}}|0\rangle_{\mathbf{Y}} \in S_{\geq t}$, we have*[17]

$$\|\,|1\rangle\langle 1|_{\mathbf{B}} U_{\mathsf{amp},T}|\phi\rangle_{\mathbf{X}}|0\rangle_{\mathbf{Y}}|0\rangle_{\mathbf{B},\mathbf{Anc}}\|^2 \geq 1-\nu.$$

   (c) *($S_{<t}$ **and** $S_{\geq t}$ **are invariant under** $U_{\mathsf{amp},T}$). For any quantum state $|\psi_{<t}\rangle_{\mathbf{X},\mathbf{Y}} \in S_{<t}$ and any b, anc, we can write*

$$U_{\mathsf{amp},T}|\psi_{<t}\rangle_{\mathbf{X},\mathbf{Y}}|b,anc\rangle_{\mathbf{B},\mathbf{Anc}} = \sum_{b',anc'} |\psi'_{<t,b',anc'}\rangle_{\mathbf{X},\mathbf{Y}}|b',anc'\rangle_{\mathbf{B},\mathbf{Anc}}$$

   by using sub-normalized states $|\psi'_{<t,b',anc'}\rangle_{\mathbf{X},\mathbf{Y}} \in S_{<t}$.
   Similarly, for any quantum state $|\psi_{\geq t}\rangle_{\mathbf{X},\mathbf{Y}} \in S_{\geq t}$ and any b, anc, we can write

$$U_{\mathsf{amp},T}|\psi_{\geq t}\rangle_{\mathbf{X},\mathbf{Y}}|b,anc\rangle_{\mathbf{B},\mathbf{Anc}} = \sum_{b',anc'} |\psi'_{\geq t,b',anc'}\rangle_{\mathbf{X},\mathbf{Y}}|b',anc'\rangle_{\mathbf{B},\mathbf{Anc}}$$

   by using sub-normalized states $|\psi'_{\geq t,b',anc'}\rangle_{\mathbf{X},\mathbf{Y}} \in S_{\geq t}$.

[17] In the previous versions of this paper, we claimed that the lower bound is $1-(1-2t+2t^2)^{T-1}(1-t)$. However, this was based on a false claim that $(1-2t+2t^2)^{T-1}(1-t)$ is decreasing in $t \in [0,1]$ for any fixed $T \geq 1$.

4. *(**Efficient Implementation of** $U_{\mathsf{amp},T}$.) There exists a QPT algorithm* Amp *(whose description is independent of* Π*) that takes as input* 1^T*, a description of quantum circuit that perform a measurement* $(\Pi, I_{\mathbf{X},\mathbf{Y}} - \Pi)$*, and a state* $|\psi\rangle_{\mathbf{X},\mathbf{Y},\mathbf{B},\mathbf{Anc}}$*, and outputs* $U_{\mathsf{amp},T}|\psi\rangle_{\mathbf{X},\mathbf{Y},\mathbf{B},\mathbf{Anc}}$*. Moreover,* Amp *uses the measurement circuit for only implementing an oracle that apply unitary to write a measurement result in a designated register in* $\mathbf{Anc}$*, and it acts on* $\mathbf{X}$ *only through the oracle access.*

Since the above lemma can be proven by a similar usage of Jordan's lemma to existing works [NWZ09, CCY20], we give the proof in Appendix D.

4 Extraction Lemma

In this section, we prove our main technical lemma, which we call the *extraction lemma*. Before giving a formal statement, we give an intuitive explanation. Suppose that we have a two-stage quantum algorithm $\mathcal{A} = (\mathcal{A}_{\mathtt{com}}, \mathcal{A}_{\mathtt{open}})$ that works as follows. $\mathcal{A}_{\mathtt{com}}$ is given pp of a commitment scheme and generates a commitment com, and passes a quantum state ρ_{st} in its internal register to $\mathcal{A}_{\mathtt{open}}$. $\mathcal{A}_{\mathtt{open}}$ is given the internal state ρ_{st}, and outputs a message-randomness pair (m, r) (which is not necessarily a valid opening to com) along with a classical output out, and let ρ'_{st} be its internal state after the execution. We call a successive execution of $\mathcal{A}_{\mathtt{com}}$ and $\mathcal{A}_{\mathtt{open}}$ a real experiment. On the other hand, we consider an *extraction experiment* where an "extractor" Ext runs on input ρ_{st} in between $\mathcal{A}_{\mathtt{com}}$ and $\mathcal{A}_{\mathtt{open}}$ to "extract" a committed message m_{Ext} while generating a simulated $\mathcal{A}$'s internal state ρ_{Ext}. Then we run $\mathcal{A}_{\mathtt{open}}$ with the internal state ρ_{Ext} instead of ρ_{st} to complete the extraction experiment. Roughly, the extraction lemma claims that if the commitment scheme is strong collapse-binding (resp. statistically binding), then there exists an extractor Ext such that we have $m = m_{\mathsf{Ext}}$ with high probability and distributions of $(m, r, \mathsf{out}, \rho'_{\mathsf{st}})$ in real and extraction experiments are computationally (resp. statistically) indistinguishable *conditioned on that* (m, r) *is a valid opening to* com.

The formal statement is given below.

Definition 4.1 (Extraction Experiments)**.** *Let* $\mathsf{Com} = (\mathsf{Setup}, \mathsf{Commit})$ *be a commitment scheme with message space* $\mathcal{M}$*, randomness space* $\mathcal{R}$*, commitment space* $\mathcal{COM}$*, and a public parameter space* $\mathcal{PP}$*. Let* $\mathcal{A} = \{\mathcal{A}_{\mathtt{com},\lambda}, \mathcal{A}_{\mathtt{open},\lambda}, \rho_\lambda\}_{\lambda\in\mathbb{N}}$ *be a sequence of two-stage non-uniform QPT algorithms with the following syntax:*

$\mathcal{A}_{\mathtt{com},\lambda}(\mathsf{pp}; \rho_\lambda) \to (\mathsf{com}, \rho_{\mathsf{st}})$**:** *It takes as input* $\mathsf{pp} \in \mathcal{PP}$ *and an advice* ρ_λ*, and outputs* $\mathsf{com} \in \mathcal{COM}$ *and a quantum state* ρ_{st} *in register* $\mathbf{ST}$*.*

$\mathcal{A}_{\mathtt{open},\lambda}(\rho_{\mathsf{st}}) \to (m, r, \mathsf{out}, \rho'_{\mathsf{st}})$**:** *It takes as input a quantum state* ρ_{st} *in register* $\mathbf{ST}$*, and outputs* $m \in \mathcal{M}$*,* $r \in \mathcal{R}$*, a classical string* out*, and a quantum state* ρ'_{st} *in register* $\mathbf{ST}$*.*

Let Ext *be a QPT algorithm and* δ *be a function in* λ*. Then we define following experiments:*

$\underline{\mathsf{Exp}_{\mathsf{real}}[\mathsf{Com}, \mathcal{A}](\lambda)}$
$\mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$,
$(\mathsf{com}, \rho_{\mathsf{st}}) \xleftarrow{\$} \mathcal{A}_{\mathtt{com},\lambda}(\mathsf{pp}; \rho_\lambda)$,
$(m, r, \mathsf{out}, \rho'_{\mathsf{st}}) \xleftarrow{\$} \mathcal{A}_{\mathtt{open},\lambda}(\rho_{\mathsf{st}})$,
If $\mathsf{Commit}(\mathsf{pp}, m; r) \neq \mathsf{com}$,
  Output $\bot$
Else Output $(\mathsf{pp}, \mathsf{com}, m, r, \mathsf{out}, \rho'_{\mathsf{st}})$.

$\underline{\mathsf{Exp}_{\mathsf{ext}}[\mathsf{Com}, \mathcal{A}, \mathsf{Ext}](\lambda, \delta)}$
$\mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$,
$(\mathsf{com}, \rho_{\mathsf{st}}) \xleftarrow{\$} \mathcal{A}_{\mathtt{com},\lambda}(\mathsf{pp}; \rho_\lambda)$,
$(m_{\mathsf{Ext}}, \rho_{\mathsf{Ext}}) \xleftarrow{\$} \mathsf{Ext}(1^\lambda, 1^{\delta^{-1}}, \mathsf{pp}, \mathsf{com}, \mathcal{A}_{\mathtt{open},\lambda}, \rho_{\mathsf{st}})$,
$(m, r, \mathsf{out}, \rho'_{\mathsf{st}}) \xleftarrow{\$} \mathcal{A}_{\mathtt{open},\lambda}(\rho_{\mathsf{Ext}})$,
If $\mathsf{Commit}(\mathsf{pp}, m; r) \neq \mathsf{com} \vee m \neq m_{\mathsf{Ext}}$,
  Output $\bot$
Else Output $(\mathsf{pp}, \mathsf{com}, m, r, \mathsf{out}, \rho'_{\mathsf{st}})$.

Lemma 4.2 (Extraction Lemma)**.** *For any strong collapse-binding commitment scheme* $\mathsf{Com} = (\mathsf{Setup}, \mathsf{Commit})$*, there exists a QPT algorithm* Ext *such that for any noticeable function* $\delta(\lambda)$ *and* $\mathcal{A} = \{\mathcal{A}_{\mathsf{com},\lambda}, \mathcal{A}_{\mathsf{open},\lambda}, \rho_\lambda\}_{\lambda\in\mathbb{N}}$ *as in Definition 4.1, we have*

$$\{\mathsf{Exp}_{\mathsf{real}}[\mathsf{Com}, \mathcal{A}](\lambda)\}_{\lambda\in\mathbb{N}} \overset{comp}{\approx}_\delta \{\mathsf{Exp}_{\mathsf{ext}}[\mathsf{Com}, \mathcal{A}, \mathsf{Ext}](\lambda, \delta)\}_{\lambda\in\mathbb{N}}. \tag{3}$$

If Com *is statistically binding instead of strong collapse-binding, we have*

$$\{\mathsf{Exp}_{\mathsf{real}}[\mathsf{Com}, \mathcal{A}](\lambda)\}_{\lambda\in\mathbb{N}} \overset{stat}{\approx}_\delta \{\mathsf{Exp}_{\mathsf{ext}}[\mathsf{Com}, \mathcal{A}, \mathsf{Ext}](\lambda, \delta)\}_{\lambda\in\mathbb{N}}. \tag{4}$$

Moreover, $\mathsf{Ext}(1^\lambda, 1^{\delta^{-1}}, \mathsf{pp}, \mathsf{com}, \mathcal{A}_{\mathsf{open},\lambda}, \rho_{\mathsf{st}})$ *works in the following manner: It uses* $\mathcal{A}_{\mathsf{open},\lambda}$ *for only implementing oracles that perform unitary part of* $\mathcal{A}_{\mathsf{open},\lambda}$ *and its inverse, and acts on* $\mathbf{ST}$ *only through black-box access to the oracles. The second output* ρ_{Ext} *of* Ext *is the state in* $\mathbf{ST}$ *after the execution. We note that* Ext *may directly act on internal registers of* $\mathcal{A}_{\mathsf{open},\lambda}$ *other than* $\mathbf{ST}$*.*

4.1 Proof of Extraction Lemma (Lemma 4.2)

Proof. (of Lemma 4.2) We first prove the lemma for the case of strong collapse-binding commitments. We explain how to modify the proof to prove the lemma for statistically binding commitments at the end of the proof.

In the proof, we need to consider sequences of many objects (e.g., unitary, Hilbert space, projection, etc.) indexed by λ. For the sake of simplicity, we will often ignore the indexing by λ.

Let U_{open} be the unitary that represents $\mathcal{A}_{\mathsf{open},\lambda}$. More precisely, we define U_{open} over a Hilbert space $\mathcal{H}_{\mathbf{A}} := \mathcal{H}_{\mathbf{ST}} \otimes \mathcal{H}_{\mathbf{W}} \otimes \mathcal{H}_{\mathbf{M}} \otimes \mathcal{H}_{\mathbf{R}} \otimes \mathcal{H}_{\mathbf{Out}}$ so that $\mathcal{A}_{\mathsf{open},\lambda}$ can be described as follows:

$\mathcal{A}_{\mathsf{open},\lambda}(\rho_{\mathsf{st}})$**:** It stores a quantum state ρ_{st} in the register $\mathbf{ST}$ and initializes registers $\mathbf{W}$, $\mathbf{M}$, $\mathbf{R}$, and $\mathbf{Out}$ to be $|0\rangle_{\mathbf{W},\mathbf{M},\mathbf{R},\mathbf{Out}}$. Then it applies a unitary U_{open}, measures registers $\mathbf{M}$, $\mathbf{R}$, and $\mathbf{Out}$ in the standard basis to obtain m, r, and out, and outputs m, r, out, and a quantum state ρ'_{st} in register $\mathbf{ST}$ tracing out register $\mathbf{W}$.

For any $\mathsf{pp} \in \mathcal{PP}$ and $\mathsf{com} \in \mathcal{COM}$, we define a projection $\Pi^{\mathsf{pp},\mathsf{com}}$ over $\mathcal{H}_{\mathbf{A}}$ as

$$\Pi^{\mathsf{pp},\mathsf{com}} := U^\dagger_{\mathsf{open}} \Pi^{\mathsf{pp},\mathsf{com}}_{\mathsf{test}} U_{\mathsf{open}} \tag{5}$$

where

$$\Pi^{\mathsf{pp},\mathsf{com}}_{\mathsf{test}} := \left(\sum_{(m,r):\mathsf{Commit}(\mathsf{pp},m;r)=\mathsf{com}} (|m,r\rangle\langle m,r|)_{\mathbf{M},\mathbf{R}} \right).$$

We apply Lemma 3.3 for $\mathcal{H}_{\mathbf{X}} := \mathcal{H}_{\mathbf{ST}}$, $\mathcal{H}_{\mathbf{Y}} := \mathcal{H}_{\mathbf{W}} \otimes \mathcal{H}_{\mathbf{M}} \otimes \mathcal{H}_{\mathbf{R}} \otimes \mathcal{H}_{\mathbf{Out}}$, $\Pi := \Pi^{\mathsf{pp},\mathsf{com}}$, $t := \delta^2/6$, and $T = \mathsf{poly}(\lambda)$ is chosen in such a way that Item 3b of Lemma 3.3 holds for $\nu = t^2$. Then we have a decomposition $(S^{\mathsf{pp},\mathsf{com}}_{<t}, S^{\mathsf{pp},\mathsf{com}}_{\geq t})$ of $\mathcal{H}_{\mathbf{A}}$ and a unitary $U^{\mathsf{pp},\mathsf{com}}_{\mathsf{amp},T}$ over $\mathcal{H}_{\mathbf{X}} \otimes \mathcal{H}_{\mathbf{Y}} \otimes \mathcal{H}_{\mathbf{B}} \otimes \mathcal{H}_{\mathbf{Anc}}$ that satisfies the requirements in Lemma 3.3 where we write $\mathsf{pp}, \mathsf{com}$ in superscript to clarify the dependence on them.

Then we construct an extractor Ext as follows:

$\mathsf{Ext}(1^\lambda, 1^{\delta^{-1}}, \mathsf{pp}, \mathsf{com}, \mathcal{A}_{\mathsf{open},\lambda}, \rho_{\mathsf{st}})$**:**

1. Store a quantum state ρ_{st} in register $\mathbf{ST}$ and initialize registers $\mathbf{W}$, $\mathbf{M}$, $\mathbf{R}$, $\mathbf{Out}$, $\mathbf{B}$, and $\mathbf{Anc}$ to be all $|0\rangle$.

2. Apply $U_{\mathsf{amp},T}^{\mathsf{pp},\mathsf{com}}$ by using the algorithm Amp in Item 4 of Lemma 3.3.
3. Measure register $\mathbf{B}$ and let b be the outcome. If $b = 0$, then return $\bot$ and immediately halt.[18] Otherwise, proceed to the next step.
4. Apply U_{open}, measure registers $\mathbf{M}$ and $\mathbf{R}$ to obtain an outcome $(m_{\mathsf{Ext}}, r_{\mathsf{Ext}})$, and apply $U_{\mathsf{open}}^{\dagger}$.
5. Apply $U_{\mathsf{amp},T}^{\mathsf{pp},\mathsf{com}\dagger}$ by using the algorithm Amp in Item 4 of Lemma 3.3.
6. Measure all registers $\mathbf{W}$, $\mathbf{M}$, $\mathbf{R}$, $\mathbf{Out}$, $\mathbf{B}$, and $\mathbf{Anc}$. If the outcome is not all 0, return $\bot$. Otherwise, output m_{Ext} and the state ρ_{Ext} in the register $\mathbf{ST}$.

We say that Ext fails if it outputs $\bot$. We can see that Ext runs in QPT and satisfies the syntactic requirements noting that Amp can be implemented by black-box access to $\mathcal{A}_{\mathsf{open},\lambda}$ by the definition of $\Pi^{\mathsf{pp},\mathsf{com}}$ and Item 4 of Lemma 3.3.

For Ext as constructed above, we consider the following sequence of hybrid experiments:

$\mathsf{Exp}_{\mathsf{ext}}[\mathsf{Com}, \mathcal{A}, \mathsf{Ext}](\lambda, \delta)$**:** This is the experiment as defined in Definition 4.1.

$\mathsf{Exp}_{\mathsf{Hyb}_1}[\mathsf{Com}, \mathcal{A}, \mathsf{Ext}](\lambda, \delta)$**:** This experiment is identical to the previous one except that the experiment only checks if Ext fails (i.e., Ext returns $\bot$) instead of checking $m \neq m_{\mathsf{Ext}}$ for the decision of outputting $\bot$. More concretely, we replace "If $\mathsf{Commit}(\mathsf{pp}, m; r) \neq \mathsf{com} \lor m \neq m_{\mathsf{Ext}}$" with "If $\mathsf{Commit}(\mathsf{pp}, m; r) \neq \mathsf{com} \lor \mathsf{Ext}$ fails".

$\mathsf{Exp}_{\mathsf{Hyb}_2}[\mathsf{Com}, \mathcal{A}, \mathsf{Ext}'](\lambda, \delta)$**:** This experiment is identical to the previous one except that instead of Ext, we use Ext' that works similarly to Ext except that Step 4 is deleted and m_{Ext} is omitted from the output. We note that the experiment is well-defined since m_{Ext} is no longer used due to the modification made in the previous hybrid.

$\mathsf{Exp}_{\mathsf{real}}[\mathsf{Com}, \mathcal{A}](\lambda)$**:** This is the experiment as defined in Definition 4.1.

We prove that output distributions of each neighboring experiments are close.

Claim 4.3. *If* Com *is strong collapse-binding, we have*

$$\{\mathsf{Exp}_{\mathsf{ext}}[\mathsf{Com}, \mathcal{A}, \mathsf{Ext}](\lambda, \delta)\}_{\lambda\in\mathbb{N}} \overset{comp}{\approx} \{\mathsf{Exp}_{\mathsf{Hyb}_1}[\mathsf{Com}, \mathcal{A}, \mathsf{Ext}](\lambda, \delta)\}_{\lambda\in\mathbb{N}}.$$

Proof. For this part, we only need the computational binding property. (As remarked in Remark 3, the strong collapse-binding property implies the computational binding property.)

We can see that the difference between these two experiments may happen only when $\mathsf{Commit}(\mathsf{pp}, m; r) = \mathsf{com}$, Ext does not fail, and $m \neq m_{\mathsf{Ext}}$. We denote this event by Bad. We prove that Bad happens with a negligible probability. When Ext does not fail, we have $b = 1$ in Step 3 of Ext. When this happens, at this point, the state in the registers $\mathbf{ST}$, $\mathbf{W}$, $\mathbf{M}$, $\mathbf{R}$, $\mathbf{Out}$ is in the span of $\Pi^{\mathsf{pp},\mathsf{com}}$ by Item 3a of Lemma 3.3. When this happens, for $(m_{\mathsf{Ext}}, r_{\mathsf{Ext}})$ obtained in Step 4, we have $\mathsf{Commit}(\mathsf{pp}, m_{\mathsf{Ext}}; r_{\mathsf{Ext}}) = \mathsf{com}$ by the definition of $\Pi^{\mathsf{pp},\mathsf{com}}$. Therefore, when Bad happens, we have $\mathsf{Commit}(\mathsf{pp}, m; r) = \mathsf{Commit}(\mathsf{pp}, m_{\mathsf{Ext}}; r_{\mathsf{Ext}}) = \mathsf{com}$ and $m \neq m_{\mathsf{Ext}}$. Thus, if this happens with non-negligible probability, we can use $\mathcal{A}$ to break the computational binding of the commitment scheme. Therefore, assuming the computational binding property of Com (which follows from the strong collapse-binding), this happens with a negligible probability. □

[18] More precisely, it returns $(m_{\mathsf{Ext}}, \rho_{\mathsf{Ext}}) := (\bot, |\bot\rangle\langle\bot|)$. The same remark also applies to Step 6

Claim 4.4. *If* Com *is strong collapse-binding, we have*

$$\{\mathsf{Exp}_{\mathsf{Hyb}_1}[\mathsf{Com}, \mathcal{A}, \mathsf{Ext}](\lambda, \delta)\}_{\lambda\in\mathbb{N}} \overset{comp}{\approx} \{\mathsf{Exp}_{\mathsf{Hyb}_2}[\mathsf{Com}, \mathcal{A}, \mathsf{Ext}'](\lambda, \delta)\}_{\lambda\in\mathbb{N}}$$

Proof. As observed in the proof of Claim 4.3, if we have $b = 1$ in Step 3 of Ext, at this point, the state in the registers $\mathbf{ST}$, $\mathbf{W}$, $\mathbf{M}$, $\mathbf{R}$, $\mathbf{Out}$ is in the span of $\Pi^{\mathsf{pp},\mathsf{com}}$ by Item 3a of Lemma 3.3. This means that conditioned on that this happens, the registers $\mathbf{M}$ and $\mathbf{R}$ contain a valid opening (m, r) for com under the public parameter pp (i.e., we have $\mathsf{Commit}(\mathsf{pp}, m; r) = \mathsf{com}$) by the definition of $\Pi^{\mathsf{pp},\mathsf{com}}$. Therefore, by the strong collapse-binding property of Com, the experiment is computationally indistinguishable even if we omit the measurement of registers $\mathbf{M}$ and $\mathbf{R}$ in Step 4. If we omit the measurement, then the Step 4 just applies U_{open} followed by $U_{\mathsf{open}}^\dagger$, which is equivalent to doing nothing. Therefore, the experiment is indistinguishable even if we delete the Step 4 of Ext and the claim is proven. □

Claim 4.5. *we have*

$$\{\mathsf{Exp}_{\mathsf{Hyb}_2}[\mathsf{Com}, \mathcal{A}, \mathsf{Ext}'](\lambda, \delta)\}_{\lambda\in\mathbb{N}} \overset{stat}{\approx}_\delta \{\mathsf{Exp}_{\mathsf{real}}[\mathsf{Com}, \mathcal{A}]\}(\lambda)\}_{\lambda\in\mathbb{N}}$$

We give a proof of Claim 4.5 in next subsection. Combining Claim 4.3, 4.4, and 4.5, we obtain Eq. 3. This completes the proof of Lemma 4.2 for the strong collapse-binding case.

Statistically binding case. Here, we briefly explain how to modify the proof to prove Lemma 4.2 when Com is statistically binding instead of strong collapse-binding commitment. The construction of Ext is the same as the above strong collapse-binding case except that it only measures the register $\mathbf{M}$ and does not measure the register $\mathbf{R}$ in Step 4. Then, the rest of the proof is done similarly to the strong collapse-binding case. We explain how we can use statistical binding instead of strong collapse-binding. In the above proof, we use strong collapse-binding property for bounding the difference between $\mathsf{Exp}_{\mathsf{ext}}$ and $\mathsf{Exp}_{\mathsf{Hyb}_1}$ (Claim 4.3) and bounding the difference between $\mathsf{Exp}_{\mathsf{Hyb}_1}$ and $\mathsf{Exp}_{\mathsf{Hyb}_2}$ (Claim 4.4).

For bounding the difference between $\mathsf{Exp}_{\mathsf{ext}}$ and $\mathsf{Exp}_{\mathsf{Hyb}_1}$, we observe that for m_{Ext} extracted by Ext, there must exist r_{Ext} such that $\mathsf{Commit}(\mathsf{pp}, m_{\mathsf{Ext}}; r_{\mathsf{Ext}}) = \mathsf{com}$ by the construction of Ext (though r_{Ext} is not measured by Ext due to the modification explained above). Therefore, if $\mathsf{Commit}(\mathsf{pp}, m; r) = \mathsf{com}$, then we must have $m_{\mathsf{Ext}} = m$ assuming that pp is binding.[19] Thus, replacing the check of $m_{\mathsf{Ext}} = m$ with the check of if Ext fails does not change the experiment unless pp is not binding, which happens with negligible probability as shown in Lemma 2.3.

For bounding the difference between $\mathsf{Exp}_{\mathsf{Hyb}_1}$ and $\mathsf{Exp}_{\mathsf{Hyb}_2}$, we observe that Step 4 of Ext (with the modification that it only measures the register $\mathbf{M}$) does not collapse the state assuming that pp is binding. Therefore, we can prove the counterpart of Claim 4.4 based on statistical binding.

We note that an upper bound of the difference between $\mathsf{Exp}_{\mathsf{Hyb}_2}$ and $\mathsf{Exp}_{\mathsf{real}}$ (Claim 4.5) can be proven by the exactly same proof since we do not use security of commitment for this part as seen in next subsection. □

4.2 Proof of Claim 4.5

In this subsection, we give a proof of Claim 4.5, which was used in the proof of Lemma 4.2 in Sec. 4.1.

[19] See Defitinion 2.2 for the definition of pp being binding.

Proof. (of Claim 4.5) We prove a stronger claim that two experiments $\mathsf{Exp}_{\mathsf{real}}[\mathsf{Com},\mathcal{A}](\lambda)$ and $\mathsf{Exp}_{\mathsf{Hyb}_2}[\mathsf{Com},\mathcal{A},\mathsf{Ext}'](\lambda,\delta)$ are statistically close for any fixed pp, com and ρ_{st}. More precisely, for any fixed pp, com and ρ_{st}, we consider the following two experiments:

$\widetilde{\mathsf{Exp}}^{\mathsf{pp},\mathsf{com}}_{\mathsf{real}}[\mathsf{Com},\mathcal{A}](\lambda,\rho_{\mathsf{st}})$	$\widetilde{\mathsf{Exp}}^{\mathsf{pp},\mathsf{com}}_{\mathsf{Hyb}_2}[\mathsf{Com},\mathcal{A},\mathsf{Ext}'](\lambda,\rho_{\mathsf{st}},\delta)$
	$\rho_{\mathsf{Ext}} \xleftarrow{\$} \mathsf{Ext}'(1^\lambda, 1^{\delta^{-1}}, \mathsf{pp}, \mathsf{com}, \mathcal{A}_{\mathsf{open},\lambda}, \rho_{\mathsf{st}})$,
$(m, r, \mathsf{out}, \rho'_{\mathsf{st}}) \xleftarrow{\$} \mathcal{A}_{\mathsf{open},\lambda}(\rho_{\mathsf{st}})$,	$(m, r, \mathsf{out}, \rho'_{\mathsf{st}}) \xleftarrow{\$} \mathcal{A}_{\mathsf{open},\lambda}(\rho_{\mathsf{Ext}})$,
If $\mathsf{Commit}(\mathsf{pp}, m; r) \neq \mathsf{com}$,	*If* $\mathsf{Commit}(\mathsf{pp}, m; r) \neq \mathsf{com}$ *or* Ext' *fails*
Output $\bot$	*Output* $\bot$
Else Output $(m, r, \mathsf{out}, \rho'_{\mathsf{st}})$.	*Else Output* $(m, r, \mathsf{out}, \rho'_{\mathsf{st}})$.

Recall that Ext' is an algorithm that works similarly to Ext except that it deletes Step 4 and does not output m_{Ext} as introduced in $\mathsf{Exp}_{\mathsf{Hyb}_2}$. We prove that for any fixed pp, com, and ρ_{st}, we have

$$\{\widetilde{\mathsf{Exp}}^{\mathsf{pp},\mathsf{com}}_{\mathsf{real}}[\mathsf{Com},\mathcal{A}](\lambda,\rho_{\mathsf{st}})\}_{\lambda\in\mathbb{N}} \overset{stat}{\approx}_{\delta} \{\widetilde{\mathsf{Exp}}^{\mathsf{pp},\mathsf{com}}_{\mathsf{Hyb}_2}[\mathsf{Com},\mathcal{A},\mathsf{Ext}'](\lambda,\rho_{\mathsf{st}},\delta)\}_{\lambda\in\mathbb{N}}. \tag{6}$$

It is easy to see that if Eq. 6 holds for all pp, com and ρ_{st}, then Claim 4.5 follows by averaging over pp, com and ρ_{st}. Moreover, since any mixed state can be understood as a probability distribution over pure states, it suffices to prove Eq. 6 assuming ρ_{st} is a pure state. Since we assume it is a pure state, we denote it by $|\phi_{\mathsf{st}}\rangle_{\mathbf{ST}}$ instead of ρ_{st}. Since we fix pp and com, we omit to write pp,com in superscripts of Π, Π_{test}, $S_{<t}$, $S_{\geq t}$, and $U_{\mathsf{amp},T}$ for notational simplicity.

In the following, we denote by $\mathbf{Other}$ to mean registers $(\mathbf{W},\mathbf{M},\mathbf{R},\mathbf{Out},\mathbf{B},\mathbf{Anc})$. Let R be an operator defined as follows:

$$R := (|0\rangle\langle 0|)_{\mathbf{Other}} U^\dagger_{\mathsf{amp},T} (|1\rangle\langle 1|)_{\mathbf{B}} U_{\mathsf{amp},T}.$$

Let $\Pi_{<t}$ and $\Pi_{\geq t}$ be projections onto $S_{<t}$ and $S_{\geq t}$, respectively. To apply Lemma 3.1, we define states $|\phi_0\rangle = |\phi_{0,0}\rangle + |\phi_{0,1}\rangle$ and $|\phi_1\rangle = |\phi_{1,0}\rangle + |\phi_{1,1}\rangle$ over $(\mathbf{D},\mathbf{ST},\mathbf{Other})$ where $\mathbf{D}$ is an additional one-qubit register as follows:

$$\begin{aligned}
|\phi_0\rangle &:= |1\rangle_{\mathbf{D}} |\phi_{\mathsf{st}}\rangle_{\mathbf{ST}} |0\rangle_{\mathbf{Other}}, \\
|\phi_{0,0}\rangle &:= |1\rangle_{\mathbf{D}} \Pi_{<t} |\phi_{\mathsf{st}}\rangle_{\mathbf{ST}} |0\rangle_{\mathbf{Other}}, \\
|\phi_{0,1}\rangle &:= |1\rangle_{\mathbf{D}} \Pi_{\geq t} |\phi_{\mathsf{st}}\rangle_{\mathbf{ST}} |0\rangle_{\mathbf{Other}}, \\
|\phi_1\rangle &:= |1\rangle_{\mathbf{D}} R |\phi_{\mathsf{st}}\rangle_{\mathbf{ST}} |0\rangle_{\mathbf{Other}} + \alpha |0\rangle_{\mathbf{D}} |0\rangle_{\mathbf{ST}} |0\rangle_{\mathbf{Other}}, \\
|\phi_{1,0}\rangle &:= |1\rangle_{\mathbf{D}} R\Pi_{<t} |\phi_{\mathsf{st}}\rangle_{\mathbf{ST}} |0\rangle_{\mathbf{Other}} + \alpha |0\rangle_{\mathbf{D}} |0\rangle_{\mathbf{ST}} |0\rangle_{\mathbf{Other}}, \\
|\phi_{1,1}\rangle &:= |1\rangle_{\mathbf{D}} R\Pi_{\geq t} |\phi_{\mathsf{st}}\rangle_{\mathbf{ST}} |0\rangle_{\mathbf{Other}}
\end{aligned}$$

for $\alpha := \sqrt{1 - \| R |\phi_{\mathsf{st}}\rangle_{\mathbf{ST}} |0\rangle_{\mathbf{Other}} \|^2}$ (so that $|\phi_1\rangle$ is a normalized state). Let Π' be a projector over $(\mathbf{D},\mathbf{ST},\mathbf{Other})$ defined as

$$\Pi' := (|1\rangle\langle 1|)_{\mathbf{D}} \otimes \Pi_{\mathbf{ST},\mathbf{W},\mathbf{M},\mathbf{R},\mathbf{Out}} \otimes I_{\mathbf{B},\mathbf{Anc}}$$

where Π is as defined in Eq. 5. (Note that we are omitting the superscript pp,com here.) Let F be the quantum algorithm as in Lemma 3.1 with respect to the projection Π' as defined above. That is, F is the algorithm that takes a state over $(\mathbf{D},\mathbf{ST},\mathbf{Other})$, applies the projective measurement $(\Pi', I - \Pi')$, and outputs the resulting state if the measurement outcome is 0, i.e., the state is projected onto Π' and otherwise outputs $\bot$. Then we have

$$\mathsf{TD}(\widetilde{\mathsf{Exp}}^{\mathsf{pp},\mathsf{com}}_{\mathsf{real}}[\mathsf{Com},\mathcal{A}](\lambda,|\phi_{\mathsf{st}}\rangle_{\mathbf{ST}}), \widetilde{\mathsf{Exp}}^{\mathsf{pp},\mathsf{com}}_{\mathsf{Hyb}_2}[\mathsf{Com},\mathcal{A},\mathsf{Ext}'](\lambda,|\phi_{\mathsf{st}}\rangle_{\mathbf{ST}},\delta)) \leq \mathsf{TD}(F(|\phi_0\rangle), F(|\phi_1\rangle)) \tag{7}$$

Indeed, this can be seen by the following observation. Let G be a quantum algorithm that works as follows: If its input is $\bot$, then G outputs $\bot$. Otherwise, G parses the input as a state over $(\mathbf{D}, \mathbf{ST}, \mathbf{Other})$, applies U_{open}, measures registers $(\mathbf{M}, \mathbf{R}, \mathbf{Out})$, and outputs registers $(\mathbf{M}, \mathbf{R}, \mathbf{Out}, \mathbf{ST})$ tracing out all the other registers. Noting that $\Pi = U_{\mathsf{open}}^{\dagger}\Pi_{\mathsf{test}}U_{\mathsf{open}}$, it is easy to see that G maps $F(|\phi_0\rangle)$ and $F(|\phi_1\rangle)$ to $\widetilde{\mathsf{Exp}}^{\mathsf{pp,com}}_{\mathsf{real}}[\mathsf{Com}, \mathcal{A}](\lambda, |\phi_{\mathsf{st}}\rangle_{\mathbf{ST}})$ and $\widetilde{\mathsf{Exp}}^{\mathsf{pp,com}}_{\mathsf{Hyb}_2}[\mathsf{Com}, \mathcal{A}, \mathsf{Ext}'](\lambda, |\phi_{\mathsf{st}}\rangle_{\mathbf{ST}}, \delta))$, respectively. Thus Eq. 7 follows from monotonicity of trace distance. Thus, it suffices to prove

$$\mathsf{TD}(F(|\phi_0\rangle), F(|\phi_1\rangle)) \leq \delta.$$

To show this by using Lemma 3.1, we prove the following claim.

Claim 4.6. *The following hold:*

1. $\langle\phi_{b,0}|\,\Pi'\,|\phi_{b',1}\rangle = 0$ *for* $b, b' \in \{0,1\}$.
2. $\|\Pi'\,|\phi_{b,0}\rangle\,\|^2 \leq t$ *for* $b \in \{0,1\}$.
3. $\|\,|\phi_{1,1}\rangle - |\phi_{0,1}\rangle\,\| \leq \sqrt{\nu}$.

Proof of Claim 4.6. The first item immediately follows from the definition. The second item for $b = 0$ immediately follows from Item 2 of Lemma 3.3. To see the second item for the case of $b = 1$, we observe that $R\Pi_{<t}\,|\phi_{\mathsf{st}}\rangle_{\mathbf{ST}}\,|0\rangle_{\mathbf{Other}}$ is in the intersection of the spans of $\Pi_{<t} \otimes I_{\mathbf{B},\mathbf{Anc}}$ and $(|0\rangle\langle 0|)_{\mathbf{Other}}$ by Item 1 and 3c of Lemma 3.3 and the definition of R. Then the desired inequality follows from Item 2 of Lemma 3.3 similarly to the case of $b = 0$. To see the third item, we observe that Item 3b of Lemma 3.3 implies

$$\|(|1\rangle\langle 1|)_{\mathbf{B}}U_{\mathsf{amp},T}\Pi_{\geq t}\,|\phi_{\mathsf{st}}\rangle_{\mathbf{ST}}\,|0\rangle_{\mathbf{Other}}\,\|^2 \leq \nu.$$

Thus, Lemma 3.2 implies

$$\|\Pi_{\geq t}\,|\phi_{\mathsf{st}}\rangle_{\mathbf{ST}}\,|0\rangle_{\mathbf{Other}} - R\Pi_{\geq t}\,|\phi_{\mathsf{st}}\rangle_{\mathbf{ST}}\,|0\rangle_{\mathbf{Other}}\,\| \leq \sqrt{\nu}.$$

This immediately implies the third item. □

By Claim 4.6, we can apply Lemma 3.1 to obtain

$$\mathsf{TD}(F(|\phi_0\rangle), F(|\phi_1\rangle)) \leq \sqrt{4t + 2\sqrt{\nu}} = \delta$$

where the final inequality follows from $t = \delta^2/6$ and $\nu = t^2$. This completes the proof of Claim 4.5. □

5 Post-Quantum ϵ-Zero-Knowledge Proof

In this section, we prove the following theorem.

Theorem 5.1. *If the QLWE assumption holds, then there exists a* 5*-round post-quantum black-box* ϵ*-zero-knowledge proof for all* **NP** *languages.*

Then we generalize it to obtain the following theorem in Sec. 5.4.

Theorem 5.2. *If a collapsing hash function exists, then there exists a* 5*-round post-quantum black-box* ϵ*-zero-knowledge proof for all* **NP** *languages.*

5.1 Construction

Our construction is the same as the Golderich-Kahan protocol [GK96] except that we instantiate the verifier's commitment with a strong collapse-binding commitment and we rely on a post-quantum delayed-witness Σ-protocol. Specifically, our construction is built on the following ingredients:

- A commitment scheme $(\mathsf{CBCom.Setup}, \mathsf{CBCom.Commit})$ that is statistical hiding and strong collapse-binding with message space $\{0,1\}^\lambda$ and randomness space $\mathcal{R}$. As noted in Sec. 2.2, such a commitment scheme exists under the QLWE assumption.
- A delayed-witness Σ-protocol $(\Sigma.P_1, \Sigma.P_3, \Sigma.V)$ for an **NP** language L as defined in Definition 2.8. As noted in Sec. 2.3.2, such a protocol exists under the QLWE assumption.

Then our construction of post-quantum black-box ϵ-zero-knowledge proof is given in Figure 1.

Protocol 1

Common Input: An instance $x \in L \cap \{0,1\}^\lambda$ for security parameter $\lambda \in \mathbb{N}$.
P's Private Input: A classical witness $w \in R_L(x)$ for x.

1. **V's Commitment to Challenge:**
   (a) P computes $\mathsf{pp} \xleftarrow{\$} \mathsf{CBCom.Setup}(1^\lambda)$ and sends pp to V.
   (b) V chooses $e \xleftarrow{\$} \{0,1\}^\lambda$ and $r \xleftarrow{\$} \mathcal{R}$, computes $\mathsf{com} \xleftarrow{\$} \mathsf{CBCom.Commit}(\mathsf{pp}, e; r)$, and sends com to P.
2. **Σ-Protocol Execution:**
   (a) P generates $(a, \mathsf{st}) \xleftarrow{\$} \Sigma.P_1(x)$ and sends a to V.
   (b) V sends (e, r) to P.
   (c) P aborts if $\mathsf{CBCom.Commit}(\mathsf{pp}, e; r) \neq \mathsf{com}$.
   Otherwise, it generates $z \xleftarrow{\$} \Sigma.P_3(\mathsf{st}, w, e)$ and sends z to V.
   (d) V outputs $\Sigma.V(x, a, e, z)$.

Figure 1: Constant-Round Post-Quantum ϵ-Zero-Knowledge Proof for $L \in \mathbf{NP}$

The completeness of the protocol clearly follows from that of the underlying Σ-protocol. In Sec. 5.2 and 5.3, we prove that this protocol satisfies statistical soundness and quantum black-box ϵ-zero-knowledge. Then we obtain Theorem 5.1.

5.2 Statistical Soundness

This is essentially the same as the proof in [GK96], but we give a proof for completeness.

For $x \notin L$ an unbounded-time cheating prover P^*, we consider the following sequence of hybrids. We denote by win_i the event that P^* wins in Hyb_i.

Hyb_1**:** This is the original game. That is,

1. P^* sends pp to V.
2. V chooses $e \xleftarrow{\$} \{0,1\}^\lambda$ and $r \xleftarrow{\$} \mathcal{R}$, computes $\mathsf{com} \xleftarrow{\$} \mathsf{CBCom.Commit}(\mathsf{pp}, e; r)$, and sends com to P^*.
3. P^* sends a to V.

4. V sends (e, r) to P^*
5. P^* sends z to V.

We say that P^* wins if we have $\Sigma.V(x, a, e, z) = \top$.

Hyb_2**:** This hybrid is identical to the previous one except that in Step 4, V uniformly chooses r' such that $\mathsf{com} = \mathsf{CBCom.Commit}(\mathsf{pp}, e; r')$ and sends (e, r') to P^* instead of (e, r). We note that this procedure may be inefficient.

This is just a conceptual change and thus we have $\Pr[\mathsf{win}_1] = \Pr[\mathsf{win}_2]$.

Hyb_3**:** This hybrid is identical to the previous one except that in Step 2, V sends $\mathsf{com} \xleftarrow{\$} \mathsf{CBCom.Commit}(\mathsf{pp}, 0^\ell; r)$ and the generation of e is delayed to Step 4.

Since no information of r is given to P^* due to the modification made in Hyb_2, by the statistical hiding property of CBCom, we have $|\Pr[\mathsf{win}_3] - \Pr[\mathsf{win}_2]| = \mathsf{negl}(\lambda)$.

Now, it is easy to prove $\Pr[\mathsf{win}_3] = \mathsf{negl}(\lambda)$ by reducing it to the statistical soundness of the Σ-protocol. Namely, we consider a cheating prover $\Sigma.P^*$ against the Σ-protocol that works as follows.

1. $\Sigma.P^*$ runs P^* to get the first message pp.
2. $\Sigma.P^*$ computes $\mathsf{com} \xleftarrow{\$} \mathsf{CBCom.Commit}(\mathsf{pp}, 0^\ell; r)$, sends com to P^*, and gets the third message a. Then $\Sigma.P^*$ sends a to its own external challenger as the first message of the Σ-protocol.
3. Upon receiving a challenge e from the external challenger, $\Sigma.P^*$ uniformly chooses r' such that $\mathsf{com} = \mathsf{CBCom.Commit}(\mathsf{pp}, e; r')$, sends (e, r') to P^*, and gets the P^*'s final message z. Then $\Sigma.P^*$ sends z to the external challenger.

It is easy to see that $\Sigma.P^*$ perfectly simulates the environment in Hyb_3 for P^*. Therefore, $\Sigma.P^*$'s winning probability is equal to $\Pr[\mathsf{win}_3]$. On the other hand, by soundness of the Σ-protocol, $\Sigma.P^*$'s winning probability is $\mathsf{negl}(\lambda)$. Therefore we have $\Pr[\mathsf{win}_3] = \mathsf{negl}(\lambda)$.

Combining the above, we have $\Pr[\mathsf{win}_1] = \mathsf{negl}(\lambda)$, which means that the protocol satisfies the statistical soundness.

5.3 Quantum Black-Box ϵ-Zero-Knowledge

Structure of the Proof. A high-level structure of our proof is similar to that of [BS20]. Specifically, we first construct simulators $\mathsf{Sim}_{\mathsf{a}}$ and $\mathsf{Sim}_{\mathsf{na}}$ that simulate the "aborting case" and "non-aborting case", respectively. More precisely, $\mathsf{Sim}_{\mathsf{a}}$ correctly simulates the verifier's view if the verifier aborts and otherwise returns a failure symbol Fail and $\mathsf{Sim}_{\mathsf{na}}$ correctly simulates the verifier's view if the verifier does not abort and otherwise returns a failure symbol Fail. Then we consider a combined simulator $\mathsf{Sim}_{\mathsf{comb}}$ that runs either of $\mathsf{Sim}_{\mathsf{a}}$ or $\mathsf{Sim}_{\mathsf{na}}$ with equal probability. Then $\mathsf{Sim}_{\mathsf{comb}}$ correctly simulates the verifier's view conditioned on that the output is not Fail, and it returns Fail with probability almost $1/2$. By applying the Watrous' quantum rewinding lemma (Lemma 2.10) to $\mathsf{Sim}_{\mathsf{comb}}$, we can convert it to a full-fledged simulator.

Though the above high-level structure is similar to [BS20], the analyses of simulators $\mathsf{Sim}_{\mathsf{a}}$ and $\mathsf{Sim}_{\mathsf{na}}$ are completely different from [BS20] since we consider different protocols. While the analysis of $\mathsf{Sim}_{\mathsf{a}}$ is easy, the analysis of $\mathsf{Sim}_{\mathsf{na}}$ is a little more complicated as it requires the extraction lemma

(Lemma 4.2), which was developed in Sec. 4.

Proof of Quantum Black-Box ϵ-Zero-Knowledge. For clarity of exposition, we first show the quantum ϵ-zero-knowledge property ignoring that the simulator should be black-box. That is, we give the full description of the malicious verifier and its quantum advice as part of the simulator's input instead of only the oracle access to the verifier. At the end of the proof, we explain that the simulator is indeed black-box.

In quantum ϵ-zero-knowledge, we need to show a simulator Sim that takes an accuracy parameter $1^{\epsilon^{-1}}$ as part of its input. We assume $\epsilon(\lambda) = o(1)$ without loss of generality since the other case trivially follows from this case. Without loss of generality, we can assume that a malicious verifier V^* does not terminate the protocol before the prover aborts since it does not gain anything by declaring the termination. We say that V^* aborts if it fails to provide a valid opening (e, r) to com in Step 2b (i.e., the prover aborts in Step 2c).

First, we construct a simulator $\mathsf{Sim}_{\mathsf{comb}}$, which returns a special symbol Fail with probability roughly $1/2$ but almost correctly simulates the output of V^*_λ conditioned on that it does not return Fail. The simulator $\mathsf{Sim}_{\mathsf{comb}}$ uses simulators $\mathsf{Sim}_{\mathsf{a}}$ and $\mathsf{Sim}_{\mathsf{na}}$ as sub-protocols:

$\mathsf{Sim}_{\mathsf{comb}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$**:**

1. Choose $\mathsf{mode} \xleftarrow{\$} \{\mathsf{a}, \mathsf{na}\}$.
2. Run $\mathsf{Sim}_{\mathsf{mode}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$.
3. Output what $\mathsf{Sim}_{\mathsf{mode}}$ outputs.

$\mathsf{Sim}_{\mathsf{a}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$**:** [20]

1. Set V^*_λ's internal state to ρ_λ.
2. Compute $\mathsf{pp} \xleftarrow{\$} \mathsf{CBCom.Setup}(1^\lambda)$ and send pp to V^*_λ.
3. V^*_λ returns com.
4. Compute $(a, \mathsf{st}) \xleftarrow{\$} \Sigma.P_1(x)$ and send a to V^*_λ.
5. V^*_λ returns (e, r).
6. Return Fail and abort if $\mathsf{CBCom.Commit}(\mathsf{pp}, e; r) = \mathsf{com}$.
   Otherwise, let V^*_λ output the final output notifying that the prover aborts.
7. The final output of V^*_λ is treated as the output $\mathsf{Sim}_{\mathsf{a}}$.

$\mathsf{Sim}_{\mathsf{na}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$**:**

1. Set V^*_λ's internal state to ρ_λ.
2. Compute $\mathsf{pp} \xleftarrow{\$} \mathsf{CBCom.Setup}(1^\lambda)$ and send pp to V^*_λ.
3. V^*_λ returns com. Let ρ_{st} be the internal state of V^*_λ at this point.
4. Compute $(e_{\mathsf{Ext}}, \rho_{\mathsf{Ext}}) \xleftarrow{\$} \mathsf{Ext}(1^\lambda, 1^{\delta^{-1}}, \mathsf{pp}, \mathsf{com}, \mathcal{A}_{\mathsf{open},\lambda}, \rho_{\mathsf{st}})$ where Ext is as in Lemma 4.2 for the commitment scheme CBCom, $\delta := \frac{\epsilon^2}{3600 \log^4(\lambda)}$, and $\mathcal{A} = (\mathcal{A}_{\mathsf{com},\lambda}, \mathcal{A}_{\mathsf{open},\lambda})$ as defined below:

   $\mathcal{A}_{\mathsf{com},\lambda}(\mathsf{pp}; \rho_\lambda)$**:** It sets V^*_λ's internal state to ρ_λ and sends pp to V^*_λ. Let com be the response by V^*_λ and ρ_{st} be the internal state of V^*_λ at this point. It outputs $(\mathsf{com}, \rho_{\mathsf{st}})$.

[20] Though $\mathsf{Sim}_{\mathsf{a}}$ does not depend on ϵ, we include $1^{\epsilon^{-1}}$ in the input for notational uniformity.

$\mathcal{A}_{\mathsf{open},\lambda}(\rho_{\mathsf{st}})$**:** It generates $(a,\mathsf{st}) \xleftarrow{\$} \Sigma.P_1(x)$,[21] sets V_λ^*'s internal state to ρ_{st}, and sends a to V_λ^*. Let (e,r) be the response by V_λ^* and let ρ'_{st} be the internal state of V_λ^* at this point. It outputs $(e,r,\mathsf{out} := (a,\mathsf{st}),\rho'_{\mathsf{st}})$.

Here, we remark that V_λ^*'s internal register corresponds to **ST** and e corresponds to m in the notation of Lemma 4.2.

5. Set the verifier's internal state to ρ_{Ext}.
6. Compute $(a,z) \xleftarrow{\$} \mathsf{Sim}_\Sigma(x,e_{\mathsf{Ext}})$ and send a to V_λ^*.
7. V_λ^* returns (e,r).
8. Return Fail and abort if $e \neq e_{\mathsf{Ext}}$ or $\mathsf{CBCom.Commit}(\mathsf{pp},e;r) \neq \mathsf{com}$. Otherwise, send z to V_λ^*.
9. The final output of V_λ^* is treated as the output $\mathsf{Sim}_{\mathsf{na}}$.

Intuitively, $\mathsf{Sim}_{\mathsf{a}}$ (resp. $\mathsf{Sim}_{\mathsf{na}}$) is a simulator that simulates the verifier's view in the case that verifier aborts (resp. does not abort).

More formally, we prove the following lemmas.

Lemma 5.3 ($\mathsf{Sim}_{\mathsf{a}}$ simulates the aborting case.)**.** *For any non-uniform QPT malicious verifier $V^* = \{V_\lambda^*,\rho_\lambda\}_{\lambda\in\mathbb{N}}$, let $\mathsf{OUT}_{V_{\mathsf{a}}^*}\langle P(w),V_\lambda^*(\rho_\lambda)\rangle(x)$ be the V_λ^*'s final output that is replaced with Fail if V_λ^* does not abort. Then we have*

$$\{\mathsf{OUT}_{V_{\mathsf{a}}^*}\langle P(w),V_\lambda^*(\rho_\lambda)\rangle(x)\}_{\lambda,x,w} \equiv \{\mathsf{Sim}_{\mathsf{a}}(x,1^{\epsilon^{-1}},V_\lambda^*,\rho_\lambda)\}_{\lambda,x,w}.$$

where $\lambda\in\mathbb{N}$, $x\in L\cap\{0,1\}^\lambda$, and $w\in R_L(x)$.

Proof. Since $\mathsf{Sim}_{\mathsf{a}}$ perfectly simulates the real execution for V_λ^* when it aborts, Lemma 5.3 immediately follows. □

Lemma 5.4 ($\mathsf{Sim}_{\mathsf{na}}$ simulates the non-aborting case.)**.** *For any non-uniform QPT malicious verifier $V^* = \{V_\lambda^*,\rho_\lambda\}_{\lambda\in\mathbb{N}}$, let $\mathsf{OUT}_{V_{\mathsf{na}}^*}\langle P(w),V_\lambda^*(\rho_\lambda)\rangle(x)$ be the V_λ^*'s final output that is replaced with Fail if V_λ^* aborts. Then we have*

$$\{\mathsf{OUT}_{V_{\mathsf{na}}^*}\langle P(w),V_\lambda^*(\rho_\lambda)\rangle(x)\}_{\lambda,x,w} \overset{comp}{\approx}_\delta \{\mathsf{Sim}_{\mathsf{na}}(x,1^{\epsilon^{-1}},V_\lambda^*,\rho_\lambda)\}_{\lambda,x,w}$$

where $\lambda\in\mathbb{N}$, $x\in L\cap\{0,1\}^\lambda$, and $w\in R_L(x)$.

Proof. Here, we analyze $\mathsf{Sim}_{\mathsf{na}}(x,1^{\epsilon^{-1}},V_\lambda^*,\rho_\lambda)$. In the following, we consider hybrid simulators $\mathsf{Sim}_{\mathsf{na},i}(x,w,1^{\epsilon^{-1}},V_\lambda^*,\rho_\lambda)$ for $i=1,2,3$. We remark that they also take the witness w as input unlike $\mathsf{Sim}_{\mathsf{na}}$.

$\mathsf{Sim}_{\mathsf{na},1}(x,w,1^{\epsilon^{-1}},V_\lambda^*,\rho_\lambda)$**:** This simulator works similarly to $\mathsf{Sim}_{\mathsf{na}}(x,1^{\epsilon^{-1}},V_\lambda^*,\rho_\lambda)$ except that it generates $(a,\mathsf{st}) \xleftarrow{\$} \Sigma.P_1(x)$ and $z \xleftarrow{\$} \Sigma.P_3(\mathsf{st},w,e_{\mathsf{Ext}})$ instead of $(a,z) \xleftarrow{\$} \mathsf{Sim}_\Sigma(x,e_{\mathsf{Ext}})$ in Step 6.

By the special honest-verifier zero-knowledge property of the Σ-protocol, we have

$$\{\mathsf{Sim}_{\mathsf{na}}(x,1^{\epsilon^{-1}},V_\lambda^*,\rho_\lambda)\}_{\lambda,x,w} \overset{comp}{\approx} \{\{\mathsf{Sim}_{\mathsf{na},1}(x,w,1^{\epsilon^{-1}},V_\lambda^*,\rho_\lambda)\}_{\lambda,x,w}\}_{\lambda,x,w}$$

where $\lambda\in\mathbb{N}$, $x\in L\cap\{0,1\}^\lambda$, and $w\in R_L(x)$.

[21] We note that we consider x to be hardwired into $\mathcal{A}_{\mathsf{open},\lambda}$. We also note that though $\mathcal{A}_{\mathsf{open},\lambda}$ does not take explicit randomness, it can generate randomness by say, applying Hadamard on its working register and then measuring it.

$\mathsf{Sim}_{\mathsf{na},2}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$**:** This simulator works similarly to $\mathsf{Sim}_{\mathsf{na},1}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$ except that the generation of z is delayed until Step 8 and it is generated as $z \xleftarrow{\$} \Sigma.P_3(\mathsf{st}, w, e)$ instead of $z \xleftarrow{\$} \Sigma.P_3(\mathsf{st}, w, e_{\mathsf{Ext}})$.

The modification does not affect the output distribution since it outputs Fail if $e \neq e_{\mathsf{Ext}}$ and if $e = e_{\mathsf{Ext}}$, then this simulator works in exactly the same way as the previous one. Therefore we have

$$\{\mathsf{Sim}_{\mathsf{na},1}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)\}_{\lambda,x,w} \equiv \{\mathsf{Sim}_{\mathsf{na},2}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)\}_{\lambda,x,w}$$

where $\lambda \in \mathbb{N}$, $x \in L \cap \{0,1\}^\lambda$, and $w \in R_L(x)$.

$\mathsf{Sim}_{\mathsf{na},3}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$**:** This simulator works similarly to $\mathsf{Sim}_{\mathsf{na},2}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$ except that Step 4 and 5 are deleted and the check of $e \neq e_{\mathsf{Ext}}$ in Step 8 is omitted. That is, it outputs Fail in Step 8 if and only if we have $\mathsf{CBCom.Commit}(\mathsf{pp}, e; r) \neq \mathsf{com}$. We note that e_{Ext} and ρ_{Ext} are no longer used at all and thus need not be generated.

We can see that Step 3 is exactly the same as executing $(\mathsf{com}, \rho_{\mathsf{st}}) \xleftarrow{\$} \mathcal{A}_{\mathsf{com},\lambda}(\mathsf{pp}; \rho_\lambda)$ and Step 6 and 7 of previous and this experiments are exactly the same as executing $(e, r, \mathsf{out} = (a, \mathsf{st}), \rho'_{\mathsf{st}}) \xleftarrow{\$} \mathcal{A}_{\mathsf{open},\lambda}(\rho_{\mathsf{Ext}})$ and $(e, r, \mathsf{out} = (a, \mathsf{st}), \rho'_{\mathsf{st}}) \xleftarrow{\$} \mathcal{A}_{\mathsf{open},\lambda}(\rho_{\mathsf{st}})$, respectively where we define ρ'_{st} in simulated experiments as V^*_λ's internal state after Step 7. Moreover, the rest of execution of the simulators can be done given $(\mathsf{pp}, \mathsf{com}, e, r, \mathsf{out} = (a, \mathsf{st}), \rho'_{\mathsf{st}})$. Therefore, by a straightforward reduction to Lemma 4.2, we have

$$\{\mathsf{Sim}_{\mathsf{na},2}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)\}_{\lambda,x,w} \overset{comp}{\approx}_\delta \{\mathsf{Sim}_{\mathsf{na},3}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)\}_{\lambda,x,w}$$

where $\lambda \in \mathbb{N}$, $x \in L \cap \{0,1\}^\lambda$, and $w \in R_L(x)$.

We can see that $\mathsf{Sim}_{\mathsf{na},3}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$ perfectly simulates the real execution for V^*_λ and outputs V^*_λ's output conditioned on that V^*_λ does not abort, and just outputs Fail otherwise. Therefore, we have

$$\{\mathsf{Sim}_{\mathsf{na},3}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)\}_{\lambda,x,w} \equiv \{\mathsf{OUT}_{V^*_{\mathsf{na}}}\langle P(w), V^*_\lambda(\rho_\lambda)\rangle(x)\}_{\lambda,x,w}$$

where $\lambda \in \mathbb{N}$, $x \in L \cap \{0,1\}^\lambda$, and $w \in R_L(x)$. Combining the above, Lemma 5.4 is proven. □

By combining Lemmas 5.3 and 5.4, we can prove the following lemma.

Lemma 5.5 ($\mathsf{Sim}_{\mathsf{comb}}$ simulates V^*_λ's output with probability almost $1/2$)**.** *For any non-uniform QPT malicious verifier $V^* = \{V^*_\lambda, \rho_\lambda\}_{\lambda\in\mathbb{N}}$, let $p^{\mathsf{suc}}_{\mathsf{comb}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$ be the probability that $\mathsf{Sim}_{\mathsf{comb}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$ does not return Fail and $D_{\mathsf{sim},\mathsf{comb}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$ be a conditional distribution of $\mathsf{Sim}_{\mathsf{comb}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$, conditioned on that it does not return Fail. There exists a negligible function negl such that for any $x = \{x_\lambda \in L \cap \{0,1\}^\lambda\}_{\lambda\in\mathbb{N}}$, we have*

$$\left|p^{\mathsf{suc}}_{\mathsf{comb}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda) - 1/2\right| \leq \delta/2 + \mathsf{negl}(\lambda). \tag{8}$$

Moreover, we have

$$\{\mathsf{OUT}_{V^*}\langle P(w), V^*_\lambda(\rho_\lambda)\rangle(x)\}_{\lambda,x,w} \overset{comp}{\approx}_{4\delta} \{D_{\mathsf{sim},\mathsf{comb}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)\}_{\lambda,x,w} \tag{9}$$

where $\lambda \in \mathbb{N}$, $x \in L \cap \{0,1\}^\lambda$, and $w \in R_L(x)$.

Proof. (sketch.) Intuition of the proof is very easy: By Lemma 5.3 and 5.4, $\mathsf{Sim}_{\mathsf{a}}$ and $\mathsf{Sim}_{\mathsf{na}}$ almost simulate the real output distribution of V^*_λ conditioned on that V^*_λ aborts and does not abort, respectively. Therefore, if we randomly guess if V^*_λ aborts and runs either of $\mathsf{Sim}_{\mathsf{a}}$ and $\mathsf{Sim}_{\mathsf{na}}$ that successfully works for the guessed case, the output distribution is close to the real output distribution of V^*_λ conditioned on that the guess is correct, which happens with probability almost $1/2$.

Indeed, the actual proof is based on the above idea, but for obtaining concrete bounds as in Eq. 8 and 9, we need some tedious calculations. We give a full proof in Appendix E since the proof is easy and very similar to that in [BS20] (once we obtain Lemma 5.3 and 5.4). □

Then, we convert $\mathsf{Sim}_{\mathsf{comb}}$ to a full-fledged simulator that does not return Fail by using the quantum rewinding lemma (Lemma 2.10). Namely, we let Q be a quantum algorithm that takes ρ_λ as input and outputs $\mathsf{Sim}_{\mathsf{comb}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$ where $b := 0$ if and only if it does not return Fail, $p_0 := \frac{1}{4}$, $q := \frac{1}{2}$, $\gamma := \delta$, and $T := 2\log(1/\delta)$. Then it is easy to check that the conditions for Lemma 2.10 is satisfied by Eq. 8 in Lemma 5.5 (for sufficiently large λ). Then by using Lemma 2.10, we can see that $\mathsf{R}(1^T, \mathsf{Q}, \rho_\lambda)$ runs in time $T \cdot |\mathsf{Q}| = \mathsf{poly}(\lambda)$ and its output (seen as a mixed state) has a trace distance bounded by $4\sqrt{\gamma}\frac{\log(1/\gamma)}{p_0(1-p_0)}$ from $D_{\mathsf{sim},\mathsf{comb}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$. Since we have $\gamma = \delta = \frac{\epsilon^2}{3600\log^4(\lambda)} = 1/\mathsf{poly}(\lambda)$, we have $4\sqrt{\gamma}\frac{\log(1/\gamma)}{p_0(1-p_0)} < 30\sqrt{\gamma}\log^2(\lambda) = \frac{\epsilon}{2}$ for sufficiently large λ where we used $\log(1/\gamma) = \log(\mathsf{poly}(\lambda)) = o(\log^2(\lambda))$. Thus, by combining the above and Eq. 9 in Lemma 5.5, if we define $\mathsf{Sim}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda) := \mathsf{R}(1^T, \mathsf{Q}, \rho_\lambda)$, then we have

$$\mathsf{OUT}_{V^*}\langle P(w), V^*_\lambda(\rho_\lambda)\rangle(x) \overset{comp}{\approx}_{\frac{\epsilon}{2}+4\delta} \mathsf{Sim}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda).$$

We can conclude the proof of quantum ϵ-zero-knowledge by noting that we have $\frac{\epsilon}{2} + 4\delta < \epsilon$ since we have $\delta = \frac{\epsilon^2}{3600\log^4(\lambda)} < \frac{\epsilon}{8}$.

Black-Box Simulation. Here, we explain that the simulator Sim constructed as above only needs black-box access to the verifier. What we need to show are that Sim applies the unitary part $U_{V^*_\lambda}$ of V^*_λ and its inverse $U^\dagger_{V^*_\lambda}$ only as oracles and Sim does not directly act on V^*_λ's internal register. There are two parts of the construction of Sim that are not obviously black-box. The first is Step 4 and 5 of $\mathsf{Sim}_{\mathsf{na}}$ where it runs the extraction algorithm Ext of Lemma 4.2, and the second is the conversion from $\mathsf{Sim}_{\mathsf{comb}}$ to Sim using R in Lemma 2.10. In the following, we explain that both steps can be implemented by black-box access to the verifier.

1. By Lemma 4.2, Ext uses the unitary part of $\mathcal{A}_{\mathsf{open},\lambda}$ and its inverse only in a black-box manner, and they can be implemented by black-box access to $U_{V^*_\lambda}$ and $U^\dagger_{V^*_\lambda}$. Moreover, since register $\mathbf{ST}$ in the notation of Lemma 4.2 corresponds to the internal register of V^*_λ in our context, the lemma ensures that Ext does not directly act on it. Also, $\mathsf{Sim}_{\mathsf{na}}$ need not explicitly set V^*_λ's internal register to ρ_{Ext} in Step 5 if we do the above black-box simulation since a state in the register automatically becomes ρ_{Ext} after the execution as stated in Lemma 4.2. Therefore, this step can be implemented by black-box access to V^*_λ.

2. Given the above observation, we now know that both $\mathsf{Sim}_{\mathsf{a}}$ and $\mathsf{Sim}_{\mathsf{na}}$ only need black-box access to V^*_λ. This means that Q only needs black-box access to V^*_λ. Since R only uses Q as oracles that perform the unitary part of Q and its inverse as stated in Lemma 2.10 and they can be implemented by black-box access to V^*_λ, R uses $U_{V^*_\lambda}$ and $U^\dagger_{V^*_\lambda}$ only as oracles. Moreover, since the register Inp in Lemma 2.10 corresponds to the internal register of V^*_λ in our context, R does not directly act on it.

By the above observations, we can see that the simulator Sim only needs black-box access to V^*_λ.

5.4 Instantiation from Collapsing Hash Function

Our construction in Figure 1 is based on two building blocks: a statistically hiding and strong collapse-binding commitment scheme and a delayed-witness Σ-protocol. Though the former can be instantiated by a collapsing hash function, we do not know how to instantiate the latter by a collapsing hash function since it needs non-interactive commitment that is not known to be implied by collapsing hash functions. However, we can just use a 4-round version of a delayed-witness Σ-protocol where the first message "commitment" in the Σ-protocol is instantiated based on Naor's commitments [Nao91] instead of a non-interactive one. Since Naor's commitments can be instantiated under any OWF and collapsing hash function is trivially also one-way, we can instantiate the 4-round version of a delayed-witness Σ-protocol based on a collapsing hash function. We can prove security of the construction based on 4-round version of a delayed-witness Σ-protocol in essentially the same manner as the security proofs in Sec. 5.2 and 5.3. We also note that this does not increase the number of rounds of our construction. Based on these observations, we obtain Theorem 5.2.

6 Post-Quantum ϵ-Zero-Knowledge Argument from OWF

In this section, we construct a constant-round ϵ-zero-knowledge argument from any post-quantum OWF.

Theorem 6.1. *If post-quantunm OWF exists, then there exists a 9-round post-quantum black-box ϵ-zero-knowledge argument for all* **NP** *languages.*

6.1 Preparation

Before giving our construction, we prepare a formalization of a variant of Blum's Graph Hamiltonicity protocol [Blu86] by Pass and Wee [PW09]. For clarity of exposition, we describe the construction and its properties in an abstracted form that is sufficient for our purpose.

Definition 6.2 (Modified Hamiltonicity Protocol)**.** *Let $(\Sigma.\mathsf{Setup}, \Sigma.\mathsf{Commit})$ be a statistically binding and computationally hiding commitment scheme with message space $\mathcal{M}_\Sigma$ randomness space $\mathcal{R}_\Sigma$, and public parameter space $\mathcal{PP}_\Sigma$. A modified Hamiltonicity protocol for an* **NP** *language L instantiated with $(\Sigma.\mathsf{Setup}, \Sigma.\mathsf{Commit})$ is a 4-round interactive proof for* **NP** *with the following syntax.*
Common Input: *An instance $x \in L \cap \{0,1\}^\lambda$ for security parameter $\lambda \in \mathbb{N}$.*
P's Private Input: *A classical witness $w \in R_L(x)$ for x.*

1. *V generates $\mathsf{pp}_\Sigma \xleftarrow{\$} \Sigma.\mathsf{Setup}(1^\lambda)$ and sends pp_Σ to P.*
2. *P generates $\{m_i\}_{i\in[\lambda]} \in \mathcal{M}_\Sigma^\lambda$ by using x (without using w). We denote this procedure by $\{m_i\}_{i\in[\lambda]} \xleftarrow{\$} \Sigma.\mathsf{Samp}(x)$. Then P picks $r_i \xleftarrow{\$} \mathcal{R}_\Sigma$ and computes $\mathsf{com}_i := \Sigma.\mathsf{Commit}(\mathsf{pp}_\Sigma, m_i; r_i)$ for all $i \in [\lambda]$ and sets $a := \{\mathsf{com}_i\}_{i\in[\lambda]}$ and $\mathsf{st} := \{m_i, r_i\}_{i\in[\lambda]}$. Then it sends a to V, and keeps st as a state information.*
3. *V chooses a "challenge" $e \xleftarrow{\$} \{0,1\}^\lambda$ and sends e to P.*
4. *P generates a "response" z from st, witness w, and e. We denote this procedure by $z \xleftarrow{\$} \Sigma.P_{\mathsf{resp}}(\mathsf{st}, w, e)$. Then it sends z to V.*

5. *V verifies the transcript $(\mathsf{pp}_\Sigma, a, e, z)$ and outputs $\top$ indicating acceptance or $\bot$ indicating rejection. We denote this procedure by $\top/\bot \xleftarrow{\$} \Sigma.V(x, \mathsf{pp}_\Sigma, a, e, z)$.*

We require the protocol to satisfy the following properties (in addition to perfect completeness and statistical soundness).

Special Honest-Verifier Zero-Knowledge. *There exist PPT simulators* $\mathsf{SimSamp}$ *and* $\mathsf{Sim}_{\mathtt{resp}}$ *such that we have*

$$\left\{ \left(\{\mathsf{com}_i\}_{i\in[\lambda]}, z\right) : \begin{array}{l} \{m_i\}_{i\in\lambda} \xleftarrow{\$} \Sigma.\mathsf{Samp}(x), \\ r_i \xleftarrow{\$} \mathcal{R}_\Sigma \text{ for } i \in [\lambda], \\ \mathsf{com}_i \xleftarrow{\$} \Sigma.\mathsf{Commit}(\mathsf{pp}_\Sigma, m_i, r_i) \text{ for } i \in [\lambda], \\ z \xleftarrow{\$} \Sigma.P_{\mathtt{resp}}(\{m_i, r_i\}, w, e) \end{array} \right\}_{\lambda,x,w,e,\mathsf{pp}_\Sigma}$$

$$\overset{comp}{\approx} \left\{ \left(\{\mathsf{com}_i\}_{i\in[\lambda]}, z\right) : \begin{array}{l} \{m_i\}_{i\in\lambda} \xleftarrow{\$} \mathsf{SimSamp}(x, e), \\ r_i \xleftarrow{\$} \mathcal{R}_\Sigma \text{ for } i \in [\lambda], \\ \mathsf{com}_i \xleftarrow{\$} \Sigma.\mathsf{Commit}(\mathsf{pp}_\Sigma, m_i, r_i) \text{ for } i \in [\lambda], \\ z \xleftarrow{\$} \mathsf{Sim}_{\mathtt{resp}}(\{m_i, r_i\}, e) \end{array} \right\}_{\lambda,x,w,e,\mathsf{pp}_\Sigma}$$

where $x \in L \cap \{0,1\}^\lambda$, $w \in R_L(x)$, $e \in \{0,1\}^\lambda$, and $\mathsf{pp}_\Sigma \in \mathcal{PP}_\Sigma$.

Bad Challenge Searchability. *Intuitively, this property requires that for any $x \notin L$ and a, if one is given a decommitment of a, one can efficiently compute a "bad challenge" e for which there may exist a valid response z. Note that we do not require that a valid response exists for the bad challenge. Rather, we require that a valid response can exist only for the bad challenge if it exists at all. A formal definition is given below.*

There exists an efficiently computable function $f_{\mathsf{bad}} : \mathcal{M}_\Sigma^\lambda \to \{0,1\}^\lambda$ such that for any binding pp_Σ,[22] $x \in \{0,1\}^\lambda \setminus L$, $a = \{\mathsf{com}_i = \Sigma.\mathsf{Commit}(\mathsf{pp}_\Sigma, m_i; r_i)\}_{i\in[\lambda]}$, $e \in \{0,1\}^\lambda \setminus \{f_{\mathsf{bad}}(\{m_i\}_{i\in[\lambda]})\}$, and z, we have $\Sigma.V(x, \mathsf{pp}_\Sigma, a, e, z) = \bot$.

Remark 6. *Looking ahead, bad challenge searchability is needed for the reduction from computational soundness of our protocol to computational hiding of a commitment scheme.*

Instantiations. The above definition is an abstraction of the modified version of Blum's Graph Hamiltonicity protocol by Pass and Wee [PW09], and this construction only requires the existence of OWF. For completeness, we briefly explain more details.

First, we recall (unparallelized version of) Blum's Graph Hamiltonicity protocol [Blu86]. In the protocol, a prover is going to prove that a graph G has a cycle where the prover is given a cycle w as a witness. In the first round, the prover picks a random permutation π, and commits to $\pi(G)$. In the second round, the verifier returns a challenge $e \in \{0,1\}$. In the third round, if $e = 0$, the prover opens all commitments and sends π, and if $e = 1$, the prover only opens commitments corresponding to the cycle $\pi(w)$. The verifier verifies the response in an obvious way. If we just implement the parallel version of Blum's Graph Hamiltonicity protocol by using (bit-wise) statistically binding commitments (e.g., Naor's commitments [Nao91]), then we can see that it already satisfies the syntactic requirements and the special honest-verifier zero-knowledge property. However, it does not seem to satisfy bad challenge searchability. The reason is that, even

[22] See definition 2.1 for the definition of binding public parameters.

if one is given a decommitment G' of the commitment, there is no efficient way to check if G and G' are isomorphic, and thus one cannot know for which challenge bit a cheating prover may answer correctly. To resolve this issue, the idea of Pass and Wee [PW09] is to let the prover commit not only to $\pi(G)$, but also to π. In this case, for G that does not have a cycle, if a decommitment is of the form $(\pi(G), \pi)$ for some permutation π, there does not exist a valid response for $e = 1$ since $\pi(G)$ does not have a cycle, and otherwise there clearly does not exist a valid response for $e = 0$. By instantiating the parallel repetition of this construction with a Naor's commitment, we obtain a protocol that satisfies the above requirements under the existence of OWF.

6.2 Construction

Our construction is inspired by that of [*PW*09], but it is not a simple instantiation of their construction since they rely on extractable commitments whose quantum security is unclear.[23] Our idea is to replace extractable commitments in their construction with usual commitment combined with witness indistinguishable proof of knowledge.

Our construction is built on the following ingredients:

- A commitment scheme $(\mathsf{SBCom.Setup}, \mathsf{SBCom.Commit})$ that is computationally hiding and statistically binding with message space $\{0,1\}^\lambda$ and randomness space $\mathcal{R}$. As noted in Sec. 2.2, such a commitment scheme exists under the existence of post-quantum OWF.
- A 4-round witness indistinguishable proof of knowledge $(\mathsf{WIPoK}.P, \mathsf{WIPoK}.V)$ for an **NP** language $\widetilde{L}$ described in Figure 2. As noted in Sec. 2.3.1, this exists under the existence of post-quantum OWF.
- Modified Hamiltonicity protocol for an **NP** language L instantiated with a computationally hiding and statistically binding commitment scheme $(\Sigma.\mathsf{Setup}, \Sigma.\mathsf{Commit})$ as defined in Definition 6.2. As discussed in Sec. 6.1, this exists under the existence of post-quantum OWF. We denote by $\mathcal{M}_\Sigma$ and $\mathcal{R}_\Sigma$ message space and randomness space of the commitment scheme.

Then our construction of post-quantum black-box ϵ-zero-knowledge argument is given in Figure 2.

The completeness of the protocol clearly follows from that of the underlying Σ-protocol. In Sec. 6.3 and 6.4, we prove that this protocol satisfies computational soundness and quantum black-box ϵ-zero-knowledge. Then we obtain Theorem 6.1.

6.3 Computational Soundness

Suppose that computational soundness does not hold. This means that there exists a non-uniform QPT adversary $P^* = \{P^*_\lambda, \rho_\lambda\}$ and a sequence of false statements $\{x \in \{0,1\}^\lambda \setminus L\}$ such that $\Pr[\mathsf{OUT}_V\langle P^*_\lambda(\rho_\lambda), V\rangle(x_\lambda) = \top]$ is non-negligible. We denote this probability by p_{win}. By an averaging argument, for at least $p_{\mathsf{win}}/2$-fraction of $(\mathsf{pp}, e, \mathsf{com}, \mathsf{pp}_\Sigma, \{\mathsf{com}_i\}_{i\in[\lambda]})$ generated in Step 1a, 1b, 2a, and 2b and P^*'s internal state ρ_{P^*} after Step 2b, we have $\Pr[\mathsf{OUT}_V\langle P^*_\lambda(\rho_\lambda), V\rangle(x_\lambda) = \top \mid (\mathsf{pp}, e, \mathsf{com}, \mathsf{pp}_\Sigma, \{\mathsf{com}_i\}_{i\in[\lambda]}, \rho_{P^*})] \geq p_{\mathsf{win}}/2$ where the above probability means a conditional probability that V returns $\top$ conditioned on $(\mathsf{pp}, e, \mathsf{com}, \mathsf{pp}_\Sigma, \{\mathsf{com}_i\}_{i\in[\lambda]}, \rho_{P^*})$. Moreover, by the statistical binding property of the commitment scheme, pp_Σ is binding except for negligible probability by Lemma 2.3. Therefore, if we define a set S consisting of $(\mathsf{pp}, e, \mathsf{com}, \mathsf{pp}_\Sigma, \{\mathsf{com}_i\}_{i\in[\lambda]}, \rho_{P^*})$ such that

[23] We could use the extractable commitment in [BS20], but that construction relies on a constant-round post-quantum zero-knowledge argument, which is stronger than our goal.

Protocol 2

Common Input: An instance $x \in L \cap \{0,1\}^\lambda$ for security parameter $\lambda \in \mathbb{N}$.
P's Private Input: A classical witness $w \in R_L(x)$ for x.

1. **V's Commitment to Challenge:**
   (a) P computes $\mathsf{pp} \xleftarrow{\$} \mathsf{SBCom.Setup}(1^\lambda)$ and sends pp to V.
   (b) V chooses $e \xleftarrow{\$} \{0,1\}^\lambda$ and $r \xleftarrow{\$} \mathcal{R}$, computes $\mathsf{com} \xleftarrow{\$} \mathsf{SBCom.Commit}(\mathsf{pp}, e; r)$, and sends com to P.
2. **First Half of Modified Hamiltonicity Protocol:**
   (a) V generates $\mathsf{pp}_\Sigma \xleftarrow{\$} \Sigma.\mathsf{Setup}(1^\lambda)$.
   (b) P generates $\{m_i\}_{i\in[\lambda]} \xleftarrow{\$} \Sigma.\mathsf{Samp}(x)$ and $\mathsf{com}_i := \Sigma.\mathsf{Commit}(\mathsf{pp}_\Sigma, m_i; r_i)$ where $r_i \xleftarrow{\$} \mathcal{R}_\Sigma$ for all $i \in [\lambda]$. It sends $a := \{\mathsf{com}_i\}_{i\in[\lambda]}$ to V and keeps $\mathsf{st} := \{m_i, r_i\}_{i\in[\lambda]}$ as a state information.
3. **Proof of Knowledge of Decommitments:** P and V interactively run the protocol $\langle \mathsf{WIPoK}.P(\{m_i, r_i\}_{i\in[\lambda]}), \mathsf{WIPoK}.V\rangle(\mathsf{pp}_\Sigma, x, a)$ where the language $\widetilde{L}$ is defined as follows:
$$\begin{aligned} &(\mathsf{pp}_\Sigma, x, a = \{\mathsf{com}_i\}_{i\in[\lambda]}) \in \widetilde{L} \\ \iff &(x \in L) \vee \\ &(\exists \{m_i, r_i\}_{i\in[\lambda]} \in (\mathcal{M}_\Sigma \times \mathcal{R}_\Sigma)^\lambda \text{ s.t. } \mathsf{com}_i = \Sigma.\mathsf{Commit}(\mathsf{pp}_\Sigma, m_i; r_i) \text{ for all } i \in [\lambda]) \end{aligned}$$
4. **Second Half of Modified Hamiltonicity Protocol:**
   (a) V sends (e, r) to P.
   (b) P aborts if $\mathsf{SBCom.Commit}(\mathsf{pp}, e; r) \neq \mathsf{com}$.
   Otherwise, it generates $z \xleftarrow{\$} \Sigma.P_{\mathsf{resp}}(\mathsf{st}, w, e)$ and sends z to V.
   (c) V outputs $\Sigma.V(x, \mathsf{pp}_\Sigma, a, e, z)$.

Figure 2: Constant-Round Post-Quantum ϵ-Zero-Knowledge Argument for $L \in \mathbf{NP}$

1. pp_Σ is binding, and
2. $\Pr[\mathsf{OUT}_V\langle P^*_\lambda(\rho_\lambda), V\rangle(x_\lambda) = \top \mid (\mathsf{pp}, e, \mathsf{com}, \mathsf{pp}_\Sigma, \{\mathsf{com}_i\}_{i\in[\lambda]}, \rho_{P^*})] \geq p_{\mathsf{win}}/2$,

then the probability that $(\mathsf{pp}, e, \mathsf{com}, \mathsf{pp}_\Sigma, \{\mathsf{com}_i\}_{i\in[\lambda]}, \rho_{P^*})$ is in S is non-negligible over the randomness of the execution of $\langle P^*_\lambda(\rho_\lambda), V\rangle(x_\lambda)$.

We fix $(\mathsf{pp}, e, \mathsf{com}, \mathsf{pp}_\Sigma, \{\mathsf{com}_i\}_{i\in[\lambda]}, \rho_{P^*}) \in S$. Since pp_Σ is binding, for each $i \in [\lambda]$, there is a unique $m_i \in \mathcal{M}$ such that $\mathsf{SBCom.Commit}(\mathsf{pp}_\Sigma, m_i; r_i) = \mathsf{com}_i$ for some $r_i \in \mathcal{R}$. By the bad challenge searchability, a valid response z can exist only for $e = f_{\mathsf{bad}}(\{m_i\}_{i\in[\lambda]})$. Therefore, for letting V accept, we must have $e = f_{\mathsf{bad}}(\{m_i\}_{i\in[\lambda]})$. Since V accepts with probability $p_{\mathsf{win}}/2 > 0$, we must have $e = f_{\mathsf{bad}}(\{m_i\}_{i\in[\lambda]})$.[24] Moreover we can use the knowledge extractor $\mathcal{K}$ of WIPoK to extract $\{m_i\}_{i\in[\lambda]}$ from P^*. That is, since the verification of WIPoK accepts with probability at least $p_{\mathsf{win}}/2$ (since otherwise the overall accepting probability should be smaller than $p_{\mathsf{win}}/2$), we have

$$\Pr[\widetilde{w} \in \mathcal{R}_{\widetilde{L}}(\mathsf{pp}_\Sigma, x_\lambda, \{\mathsf{com}_i\}_{i\in\lambda}) : \widetilde{w} \xleftarrow{\$} \mathcal{K}^{P^*_\lambda(\rho_{P^*})}(\mathsf{pp}_\Sigma, x_\lambda, \{\mathsf{com}_i\}_{i\in[\lambda]})] \geq \frac{1}{\mathsf{poly}(\lambda)} \cdot (p_{\mathsf{win}}/2)^d - \mathsf{negl}(\lambda)$$

[24] Strictly speaking, we may have $p_{\mathsf{win}}/2 = 0$ for a finite number of λ. This can be easily dealt with by considering sufficiently large λ. For simplicity, we omit this.

where one can see that the RHS is non-negligible. Since we assume $x_\lambda \notin L$ and pp_Σ is binding, when we have $\widetilde{w} \in \mathcal{R}_{\widetilde{L}}(\mathsf{pp}_\Sigma, x_\lambda, \{\mathsf{com}_i\}_{i\in\lambda})$, we have $\widetilde{w} = \{m_i, r'_i\}_{i\in[\lambda]}$ for some $\{r'_i\}_{i\in[\lambda]}$. By using this extracted witness, one can compute $e = f_{\mathsf{bad}}(\{m_i\}_{i\in[\lambda]})$. We can use this to show contradiction to the unpredictability of the commitment scheme.

Specifically, we construct an adversary $\mathcal{A} = \{\mathcal{A}_\lambda, \rho_{\mathcal{A},\lambda}\}_{\lambda\in\mathbb{N}}$ that breaks the unpredictability of SBCom, which contradicts the computational hiding property as noted in Lemma 2.5.

Advice: $\mathcal{A}$ gets an advice $\rho_{\mathcal{A},\lambda} = (x_\lambda, \rho_\lambda)$

$\mathcal{A}_\lambda(\rho_{\mathcal{A},\lambda})$**:** It sets P^*_λ's internal register to ρ_λ, gives x_λ as input to P^*_λ, receives pp from P^*_λ, and sends pp to its external challenger. Let com be challenger's response. (Here, e is implicitly chosen by the challenger.) $\mathcal{A}$ gives com to P^*_λ as a message from V, and simulates the protocol between P^*_λ and V until Step 2b. Let ρ_{P^*} be P^*_λ's internal state at this point. Then it runs $\{m^*_i, r^*_i\}_{i\in[\lambda]} \xleftarrow{\$} \mathcal{K}^{P^*_\lambda(\rho_{P^*})}(\mathsf{pp}_\Sigma, x_\lambda, \{\mathsf{com}_i\}_{i\in[\lambda]})$ computes $e^* = f_{\mathsf{bad}}(\{m^*_i\}_{i\in[\lambda]})$, and outputs e^*.

We can see that $\mathcal{A}$ perfectly simulates the soundness game until Step 2b. Therefore, we have $(\mathsf{pp}, e, \mathsf{com}, \mathsf{pp}_\Sigma, \{\mathsf{com}_i\}_{i\in[\lambda]}, \rho_{P^*}) \in S$ with non-negligible probability, and for any fixed such values, we have $e^* = e$ with non-negligible probability as observed in the previous paragraph. Therefore, $\mathcal{A}$ succeeds in finding e with non-negligible probability overall, and breaks the unpredictability.

Since this contradicts the unpredictability, which follows from the assumed computational hiding property, there does not exist non-uniform QPT P^* that breaks soundness.

6.4 Quantum Black-Box ϵ-Zero-Knowledge

The proof is similar to that in Sec. 5.3 except that we need to deal with WIPoK. Similarly to the proof there, we show the quantum ϵ-zero-knowledge property ignoring that the simulator should be black-box for clarity of exposition. One can see that the simulator is indeed black-box by similar observations made at the end of Sec. 5.3.

In quantum ϵ-zero-knowledge, we need to show a simulator Sim that takes an accuracy parameter $1^{\epsilon^{-1}}$ as part of its input. We assume $\epsilon(\lambda) = o(1)$ without loss of generality since the other case trivially follows from this case. Without loss of generality, we can assume that a malicious verifier V^* does not terminate the protocol before the prover aborts since it does not gain anything by declaring the termination. We say that V^* aborts if it fails to provide a valid opening (e, r) to com in Step 4a (i.e., the prover aborts in Step 4b).

First, we construct a simulator $\mathsf{Sim}_{\mathsf{comb}}$, which returns a special symbol Fail with probability roughly 1/2 but almost correctly simulates the output of V^*_λ conditioned on that it does not return Fail. The simulator $\mathsf{Sim}_{\mathsf{comb}}$ uses simulators $\mathsf{Sim}_{\mathsf{a}}$ and $\mathsf{Sim}_{\mathsf{na}}$ as sub-protocols:

$\mathsf{Sim}_{\mathsf{comb}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$**:**

1. Choose $\mathsf{mode} \xleftarrow{\$} \{\mathsf{a}, \mathsf{na}\}$.
2. Run $\mathsf{Sim}_{\mathsf{mode}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$.
3. Output what $\mathsf{Sim}_{\mathsf{mode}}$ outputs.

$\mathsf{Sim}_{\mathsf{a}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$**:**

1. Set V^*_λ's internal state to ρ_λ.
2. Compute $\mathsf{pp} \xleftarrow{\$} \mathsf{SBCom.Setup}(1^\lambda)$ and send pp to V^*_λ.

3. V^*_λ returns com and pp_Σ.
4. Compute $\{m_i\}_{i\in[\lambda]} \xleftarrow{\$} \Sigma.\mathsf{Samp}(x)$ and $\mathsf{com}_i := \Sigma.\mathsf{Commit}(\mathsf{pp}_\Sigma, m_i; r_i)$ where $r_i \xleftarrow{\$} \mathcal{R}_\Sigma$ for all $i \in [\lambda]$, and sends $a := \{\mathsf{com}_i\}_{i\in[\lambda]}$ to V^*_λ. Let $\rho_{V^*_\lambda}$ be V^*_λ's internal state at this point.
5. Interactively execute $\langle \mathsf{WIPoK}.P(\{m_i, r_i\}_{i\in[\lambda]}), \mathsf{WIPoK}.V^*_\lambda(\rho_{V^*_\lambda})\rangle(\mathsf{pp}_\Sigma, x, a)$ where $\mathsf{WIPoK}.V^*_\lambda$ is the corresponding part of V^*_λ.
6. V^*_λ returns (e, r).
7. Return Fail and abort if $\mathsf{SBCom.Commit}(\mathsf{pp}, e; r) = \mathsf{com}$.
   Otherwise, let V^*_λ output the final output notifying that the prover aborts.
8. The final output of V^*_λ is treated as the output $\mathsf{Sim}_{\mathsf{a}}$.

$\mathsf{Sim}_{\mathsf{na}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$**:**

1. Set V^*_λ's internal state to ρ_λ.
2. Compute $\mathsf{pp} \xleftarrow{\$} \mathsf{SBCom.Setup}(1^\lambda)$ and send pp to V^*_λ.
3. V^*_λ returns com. Let ρ_{st} be the internal state of V^*_λ at this point.[25]
4. Compute $(e_{\mathsf{Ext}}, \rho_{\mathsf{Ext}}) \xleftarrow{\$} \mathsf{Ext}(1^\lambda, 1^{\delta^{-1}}, x, \mathsf{pp}, \mathsf{com}, \mathcal{A}_{\mathtt{open},\lambda}, \rho_{\mathsf{st}})$ where Ext is as in Lemma 4.2 for the commitment scheme SBCom, $\delta := \frac{\epsilon^2}{3600\log^4(\lambda)}$, and $\mathcal{A} = (\mathcal{A}_{\mathtt{com},\lambda}, \mathcal{A}_{\mathtt{open},\lambda})$ is defined as follows:

   $\mathcal{A}_{\mathtt{com},\lambda}(\mathsf{pp}; \rho_\lambda)$**:** It sets V^*_λ's internal state to ρ_λ and sends pp to V^*_λ. Let com be the response by V^*_λ and ρ_{st} be the internal state of V^*_λ at this point. It outputs $(\mathsf{com}, \rho_{\mathsf{st}})$.

   $\mathcal{A}_{\mathtt{open},\lambda}(\rho_{\mathsf{st}})$**:** It sets V^*_λ's internal state to ρ_{st}, and receives pp_Σ from V^*_λ. It generates $\{m_i\}_{i\in[\lambda]} \xleftarrow{\$} \Sigma.\mathsf{Samp}(x)$ and $\mathsf{com}_i := \Sigma.\mathsf{Commit}(\mathsf{pp}_\Sigma, m_i; r_i)$ where $r_i \xleftarrow{\$} \mathcal{R}_\Sigma$ for all $i \in [\lambda]$ and sends $a := \{\mathsf{com}_i\}_{i\in[\lambda]}$ to V^*_λ. Let $\rho_{V^*_\lambda}$ be V^*_λ's internal state at this point. Then it executes $\langle \mathsf{WIPoK}.P(\{m_i, r_i\}_{i\in[\lambda]}), \mathsf{WIPoK}.V^*_\lambda(\rho_{V^*_\lambda})\rangle(\mathsf{pp}_\Sigma, x, a)$ where $\mathsf{WIPoK}.V^*_\lambda$ is the corresponding part of V^*_λ. After completing the execution of WIPoK, V^*_λ returns (e, r). Let ρ'_{st} be the internal state of V^*_λ at this point. It outputs $(e, r, \mathsf{out} := (a, \{m_i, r_i\}_{i\in[\lambda]}), \rho'_{\mathsf{st}})$.

   Here, we remark that V^*_λ's internal register corresponds to $\mathbf{ST}$ and e corresponds to m in the notation of Lemma 4.2.
5. Set the verifier's internal state to ρ_{Ext}.
6. V^* returns pp_Σ.
7. Compute $\{m_i\}_{i\in\lambda} \xleftarrow{\$} \mathsf{SimSamp}(x, e_{\mathsf{Ext}})$, $\mathsf{com}_i \xleftarrow{\$} \Sigma.\mathsf{Commit}(\mathsf{pp}_\Sigma, m_i, r_i)$ where $r_i \xleftarrow{\$} \mathcal{R}_\Sigma$ for all $i \in [\lambda]$, and $z \xleftarrow{\$} \mathsf{Sim}_{\mathtt{resp}}(\{m_i, r_i\}, e_{\mathsf{Ext}})$. Send $a := \{\mathsf{com}_i\}_{i\in[\lambda]}$ to V^*_λ. Let $\rho_{V^*_\lambda}$ be V^*_λ's internal state at this point.
8. Interactively execute $\langle \mathsf{WIPoK}.P(\{m_i, r_i\}_{i\in[\lambda]}), \mathsf{WIPoK}.V^*_\lambda(\rho_{V^*_\lambda})\rangle(\mathsf{pp}_\Sigma, x, a)$ where $\mathsf{WIPoK}.V^*_\lambda$ is the corresponding part of V^*_λ.
9. V^*_λ returns (e, r).
10. Return Fail and abort if $e \neq e_{\mathsf{Ext}}$ or $\mathsf{SBCom.Commit}(\mathsf{pp}, e; r) \neq \mathsf{com}$.
    Otherwise, send z to V^*_λ.

[25] Though com and pp_Σ can be sent simultaneously in the real protocol, we consider that they are sent one by one, and the state ρ_{st} is defined to be V^*_λ's internal state in between them. We stress that this is just for convenience of the proof, and the protocol satisfies the same security even if the verifier sends com and pp_Σ simultaneously.

11. The final output of V_λ^* is treated as the output $\mathsf{Sim}_{\mathsf{na}}$.

Intuitively, $\mathsf{Sim}_{\mathsf{a}}$ (resp. $\mathsf{Sim}_{\mathsf{na}}$) is a simulator that simulates the verifier's view in the case that verifier aborts (resp. does not abort).

More formally, we prove the following lemmas.

Lemma 6.3 ($\mathsf{Sim}_{\mathsf{a}}$ simulates the aborting case.)**.** *For any non-uniform QPT malicious verifier $V^* = \{V_\lambda^*, \rho_\lambda\}_{\lambda\in\mathbb{N}}$, let $\mathsf{OUT}_{V_{\mathsf{a}}^*}\langle P(w), V_\lambda^*(\rho_\lambda)\rangle(x)$ be the V_λ^*'s final output that is replaced with Fail if V_λ^* does not abort. Then we have*

$$\{\mathsf{OUT}_{V_{\mathsf{a}}^*}\langle P(w), V_\lambda^*(\rho_\lambda)\rangle(x)\}_{\lambda,x,w} \equiv \{\mathsf{Sim}_{\mathsf{a}}(x, 1^{\epsilon^{-1}}, V_\lambda^*, \rho_\lambda)\}_{\lambda,x,w}.$$

where $\lambda \in \mathbb{N}$, $x \in L \cap \{0,1\}^\lambda$, and $w \in R_L(x)$.

Proof. Since $\mathsf{Sim}_{\mathsf{a}}$ perfectly simulates the real execution for V_λ^* when it aborts, Lemma 6.3 immediately follows. □

Lemma 6.4 ($\mathsf{Sim}_{\mathsf{na}}$ simulates the non-aborting case.)**.** *For any non-uniform QPT malicious verifier $V^* = \{V_\lambda^*, \rho_\lambda\}_{\lambda\in\mathbb{N}}$, let $\mathsf{OUT}_{V_{\mathsf{na}}^*}\langle P(w), V_\lambda^*(\rho_\lambda)\rangle(x)$ be the V_λ^*'s final output that is replaced with Fail if V_λ^* aborts. Then we have*

$$\{\mathsf{OUT}_{V_{\mathsf{na}}^*}\langle P(w), V_\lambda^*(\rho_\lambda)\rangle(x)\}_{\lambda,x,w} \overset{comp}{\approx}_\delta \{\mathsf{Sim}_{\mathsf{na}}(x, 1^{\epsilon^{-1}}, V_\lambda^*, \rho_\lambda)\}_{\lambda,x,w}$$

where $\lambda \in \mathbb{N}$, $x \in L \cap \{0,1\}^\lambda$, and $w \in R_L(x)$.

Proof. Here, we analyze $\mathsf{Sim}_{\mathsf{na}}(x, 1^{\epsilon^{-1}}, V_\lambda^*, \rho_\lambda)$. In the following, we consider hybrid simulators $\mathsf{Sim}_{\mathsf{na},i}(x, w, 1^{\epsilon^{-1}}, V_\lambda^*, \rho_\lambda)$ for $i = 1, 2, 3, 4, 5$. We remark that they also take the witness w as input unlike $\mathsf{Sim}_{\mathsf{na}}$.

$\mathsf{Sim}_{\mathsf{na},1}(x, w, 1^{\epsilon^{-1}}, V_\lambda^*, \rho_\lambda)$**:** This simulator works similarly to $\mathsf{Sim}_{\mathsf{na}}(x, 1^{\epsilon^{-1}}, V_\lambda^*, \rho_\lambda)$ except that in the simulation of WIPoK in Step 8, it uses witness w instead of $\{m_i, r_i\}_{i\in[\lambda]}$.

By witness indistinguishability of WIPoK, we have

$$\{\mathsf{Sim}_{\mathsf{na}}(x, 1^{\epsilon^{-1}}, V_\lambda^*, \rho_\lambda)\}_{\lambda,x,w} \overset{comp}{\approx} \{\mathsf{Sim}_{\mathsf{na},1}(x, w, 1^{\epsilon^{-1}}, V_\lambda^*, \rho_\lambda)\}_{\lambda,x,w}$$

where $\lambda \in \mathbb{N}$, $x \in L \cap \{0,1\}^\lambda$, and $w \in R_L(x)$.

$\mathsf{Sim}_{\mathsf{na},2}(x, w, 1^{\epsilon^{-1}}, V_\lambda^*, \rho_\lambda)$**:** This simulator works similarly to $\mathsf{Sim}_{\mathsf{na},1}(x, w, 1^{\epsilon^{-1}}, V_\lambda^*, \rho_\lambda)$ except that it generates $(a = \{\mathsf{com}_i\}_{i\in[\lambda]}, z)$ as in the real protocol for the challenge e_{Ext} instead of using the simulator in Step 7. That is, it generates $\{m_i\}_{i\in[\lambda]} \overset{\$}{\leftarrow} \Sigma.\mathsf{Samp}(x)$, $\mathsf{com}_i := \Sigma.\mathsf{Commit}(\mathsf{pp}_\Sigma, m_i; r_i)$ where $r_i \overset{\$}{\leftarrow} \mathcal{R}_\Sigma$ for all $i \in [\lambda]$, and $z \overset{\$}{\leftarrow} \Sigma.P_{\mathtt{resp}}(\mathsf{st}, w, e_{\mathsf{Ext}})$ where $\mathsf{st} := \{m_i, r_i\}_{i\in[\lambda]}$.

By the special honest-verifier zero-knowledge property of the modified Hamiltonicity protocol, we have

$$\{\mathsf{Sim}_{\mathsf{na},1}(x, w, 1^{\epsilon^{-1}}, V_\lambda^*, \rho_\lambda)\}_{\lambda,x,w} \overset{comp}{\approx} \{\mathsf{Sim}_{\mathsf{na},2}(x, w, 1^{\epsilon^{-1}}, V_\lambda^*, \rho_\lambda)\}_{\lambda,x,w}$$

where $\lambda \in \mathbb{N}$, $x \in L \cap \{0,1\}^\lambda$, and $w \in R_L(x)$.

$\mathsf{Sim}_{\mathsf{na},3}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$**:** This simulator works similarly to $\mathsf{Sim}_{\mathsf{na},2}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$ except that in the simulation of WIPoK in Step 8, it uses witness $\{m_i, r_i\}_{i\in[\lambda]}$ instead of w.

By witness indistinguishability of WIPoK, we have

$$\{\mathsf{Sim}_{\mathsf{na},2}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)\}_{\lambda,x,w} \overset{comp}{\approx} \{\mathsf{Sim}_{\mathsf{na},3}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)\}_{\lambda,x,w}$$

where $\lambda \in \mathbb{N}$, $x \in L \cap \{0,1\}^\lambda$, and $w \in R_L(x)$.

$\mathsf{Sim}_{\mathsf{na},4}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$**:** This simulator works similarly to $\mathsf{Sim}_{\mathsf{na},3}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$ except that the generation of z is delayed until Step 10 and it is generated as $z \overset{\$}{\leftarrow} \Sigma.P_{\mathtt{resp}}(\mathsf{st}, w, e)$ instead of $z \overset{\$}{\leftarrow} \Sigma.P_{\mathtt{resp}}(\mathsf{st}, w, e_{\mathsf{Ext}})$.

The modification does not affect the output distribution since it outputs Fail if $e \neq e_{\mathsf{Ext}}$ and if $e = e_{\mathsf{Ext}}$, then this simulator works in exactly the same way as the previous one. Therefore we have

$$\{\mathsf{Sim}_{\mathsf{na},3}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)\}_{\lambda,x,w} \equiv \{\mathsf{Sim}_{\mathsf{na},4}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)\}_{\lambda,x,w}$$

where $\lambda \in \mathbb{N}$, $x \in L \cap \{0,1\}^\lambda$, and $w \in R_L(x)$.

$\mathsf{Sim}_{\mathsf{na},5}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$**:** This simulator works similarly to $\mathsf{Sim}_{\mathsf{na},4}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$ except that Step 4 and 5 are deleted and the check of $e \neq e_{\mathsf{Ext}}$ in Step 10 is omitted. That is, it outputs Fail in Step 10 if and only if we have $\mathsf{SBCom.Commit}(\mathsf{pp}, e; r) \neq \mathsf{com}$. We note that e_{Ext} and ρ_{Ext} are no longer used at all and thus need not be generated.

We can see that Step 3 is exactly the same as executing $(\mathsf{com}, \rho_{\mathsf{st}}) \overset{\$}{\leftarrow} \mathcal{A}_{\mathtt{com},\lambda}(\mathsf{pp}; \rho_\lambda)$ and Step 6, 7, 8, and 9 of previous and this simulators are exactly the same as executing $(e, r, \mathsf{out} = (a, \{m_i, r_i\}_{i\in[\lambda]}), \rho'_{\mathsf{st}}) \overset{\$}{\leftarrow} \mathcal{A}_{\mathsf{open},\lambda}(\rho_{\mathsf{Ext}})$ and $(e, r, \mathsf{out} = (a, \{m_i, r_i\}_{i\in[\lambda]}), \rho'_{\mathsf{st}}) \overset{\$}{\leftarrow} \mathcal{A}_{\mathsf{open},\lambda}(\rho_{\mathsf{st}})$, respectively where we define ρ'_{st} in simulated experiments as V^*_λ's internal state after Step 9. Moreover, the rest of execution of the simulators can be done given $(\mathsf{pp}, \mathsf{com}, e, r, \mathsf{out} = (a, \mathsf{st}), \rho'_{\mathsf{st}})$. Therefore, by a straightforward reduction to Lemma 4.2, we have

$$\{\mathsf{Sim}_{\mathsf{na},4}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)\}_{\lambda,x,w} \overset{stat}{\approx}_\delta \{\mathsf{Sim}_{\mathsf{na},5}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)\}_{\lambda,x,w}$$

where $\lambda \in \mathbb{N}$, $x \in L \cap \{0,1\}^\lambda$, and $w \in R_L(x)$.

We can see that $\mathsf{Sim}_{\mathsf{na},5}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$ perfectly simulates the real execution for V^*_λ and outputs V^*_λ's output conditioned on that V^*_λ does not abort, and just outputs Fail otherwise. Therefore, we have

$$\{\mathsf{Sim}_{\mathsf{na},5}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)\}_{\lambda,x,w} \equiv \{\mathsf{OUT}_{V^*_{\mathsf{na}}}\langle P(w), V^*_\lambda(\rho_\lambda)\rangle(x)\}_{\lambda,x,w}$$

where $\lambda \in \mathbb{N}$, $x \in L \cap \{0,1\}^\lambda$, and $w \in R_L(x)$. Combining the above, Lemma 6.4 is proven. □

By combining Lemmas 6.3 and 6.4, we can prove the following lemma.

Lemma 6.5 ($\mathsf{Sim}_{\mathsf{comb}}$ simulates V^*_λ's output with probability almost 1/2)**.** *For any non-uniform QPT malicious verifier $V^* = \{V^*_\lambda, \rho_\lambda\}_{\lambda\in\mathbb{N}}$, let $p^{\mathsf{suc}}_{\mathsf{comb}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$ be the probability that $\mathsf{Sim}_{\mathsf{comb}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$ does not return Fail and $D_{\mathsf{sim},\mathsf{comb}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$ be a conditional distribution of $\mathsf{Sim}_{\mathsf{comb}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$,*

conditioned on that it does not return Fail. *There exists a negligible function* negl *such that for any* $x = \{x_\lambda \in L \cap \{0,1\}^\lambda\}_{\lambda\in\mathbb{N}}$, *we have*

$$\left|p^{\mathsf{suc}}_{\mathsf{comb}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda) - 1/2\right| \leq \delta/2 + \mathsf{negl}(\lambda). \tag{10}$$

Moreover, we have

$$\{\mathsf{OUT}_{V^*}\langle P(w), V^*_\lambda(\rho_\lambda)\rangle(x)\}_{\lambda,x,w} \overset{comp}{\approx}_{4\delta} \{D_{\mathsf{sim},\mathsf{comb}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)\}_{\lambda,x,w} \tag{11}$$

where $\lambda \in \mathbb{N}$, $x \in L \cap \{0,1\}^\lambda$, *and* $w \in R_L(x)$.

Proof. The proof of this lemma is exactly the same as that of Lemma 5.5 in Appendix E. □

Finally, we convert $\mathsf{Sim}_{\mathsf{comb}}$ to a full-fledged simulator that does not return Fail by using the quantum rewinding lemma (Lemma 2.10). This part is exactly the same as that in Sec. 5.3. Finally, we can see that our simulator is black-box similarly to the last paragraph of Sec. 5.3. This completes the proof of quantum black-box ϵ-zero-knowledge property.

Acknowledgement

NHC's research is support by the U.S. Department of Defense and NIST through the Hartree Postdoctoral Fellowship at QuICS and by NSF through IUCRC Planning Grant Indiana University: Center for Quantum Technologies (CQT) under award number 2052730. KMC's research is partially supported by MOST, Taiwan, under Grant no. MOST 109-2223-E-001-001-MY3 and Executive Yuan Data Safety and Talent Cultivation Project (ASKPQ-109-DSTCP).

References

[ACGH20] Gorjan Alagic, Andrew M. Childs, Alex B. Grilo, and Shih-Han Hung. Non-interactive classical verification of quantum computation. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020, Part III*, volume 12552 of *LNCS*, pages 153–180. Springer, Heidelberg, November 2020.

[AL20] Prabhanjan Ananth and Rolando L. La Placa. Secure quantum extraction protocols. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020, Part III*, volume 12552 of *LNCS*, pages 123–152. Springer, Heidelberg, November 2020.

[AR06] N. Aharon and Oded Regev. Witness-preserving amplification of qma (lecture note), 2006. https://cims.nyu.edu/~regev/teaching/quantum_fall_2005/ln/qma.pdf.

[BC90] Gilles Brassard and Claude Crépeau. Sorting out zero-knowledge. In Jean-Jacques Quisquater and Joos Vandewalle, editors, *EUROCRYPT'89*, volume 434 of *LNCS*, pages 181–191. Springer, Heidelberg, April 1990.

[BCY91] Gilles Brassard, Claude Crépeau, and Moti Yung. Constant-round perfect zero-knowledge computationally convincing protocols. *Theor. Comput. Sci.*, 84(1):23–52, 1991.

[BG20] Anne Broadbent and Alex B. Grilo. QMA-hardness of consistency of local density matrices with applications to quantum zero-knowledge. In *61st FOCS*, pages 196–205. IEEE Computer Society Press, November 2020.

[BJSW20] Anne Broadbent, Zhengfeng Ji, Fang Song, and John Watrous. Zero-knowledge proof systems for QMA. *SIAM J. Comput.*, 49(2):245–283, 2020.

[BKP18] Nir Bitansky, Yael Tauman Kalai, and Omer Paneth. Multi-collision resistance: a paradigm for keyless hash functions. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *50th ACM STOC*, pages 671–684. ACM Press, June 2018.

[BKP19] Nir Bitansky, Dakshita Khurana, and Omer Paneth. Weak zero-knowledge beyond the black-box barrier. In Moses Charikar and Edith Cohen, editors, *51st ACM STOC*, pages 1091–1102. ACM Press, June 2019.

[BL02] Boaz Barak and Yehuda Lindell. Strict polynomial-time in simulation and extraction. In *34th ACM STOC*, pages 484–493. ACM Press, May 2002.

[BLP+13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 575–584. ACM Press, June 2013.

[Blu86] Manuel Blum. How to prove a theorem so no one else can claim it. In *Proceedings of the International Congress of Mathematicians*, page 1444–1451, 1986.

[BP12] Nir Bitansky and Omer Paneth. Point obfuscation and 3-round zero-knowledge. In Ronald Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*, pages 190–208. Springer, Heidelberg, March 2012.

[Bra18] Zvika Brakerski. Quantum FHE (almost) as secure as classical. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part III*, volume 10993 of *LNCS*, pages 67–95. Springer, Heidelberg, August 2018.

[BS20] Nir Bitansky and Omri Shmueli. Post-quantum zero knowledge in constant rounds. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *52nd ACM STOC*, pages 269–279. ACM Press, June 2020.

[BY20] Zvika Brakerski and Henry Yuen. Quantum garbled circuits. *arXiv*, 2006.01085, 2020.

[CCLY21] Nai-Hui Chia, Kai-Min Chung, Qipeng Liu, and Takashi Yamakawa. On the impossibility of post-quantum black-box zero-knowledge in constant rounds. *arXiv*, 2103.11244, 2021.

[CCY20] Nai-Hui Chia, Kai-Min Chung, and Takashi Yamakawa. Classical verification of quantum computations with efficient verifier. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020, Part III*, volume 12552 of *LNCS*, pages 181–206. Springer, Heidelberg, November 2020.

[CLP15] Kai-Min Chung, Edward Lui, and Rafael Pass. From weak to strong zero-knowledge and applications. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part I*, volume 9014 of *LNCS*, pages 66–92. Springer, Heidelberg, March 2015.

[CVZ20] Andrea Coladangelo, Thomas Vidick, and Tina Zhang. Non-interactive zero-knowledge arguments for QMA, with preprocessing. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part III*, volume 12172 of *LNCS*, pages 799–828. Springer, Heidelberg, August 2020.

[DFS04] Ivan Damgård, Serge Fehr, and Louis Salvail. Zero-knowledge proofs and string commitments withstanding quantum attacks. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 254–272. Springer, Heidelberg, August 2004.

[DNRS03] Cynthia Dwork, Moni Naor, Omer Reingold, and Larry J. Stockmeyer. Magic functions. *J. ACM*, 50(6):852–921, 2003.

[DNS04] Cynthia Dwork, Moni Naor, and Amit Sahai. Concurrent zero-knowledge. *J. ACM*, 51(6):851–898, 2004.

[FGJ18] Nils Fleischhacker, Vipul Goyal, and Abhishek Jain. On the existence of three round zero-knowledge proofs. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*, pages 3–33. Springer, Heidelberg, April / May 2018.

[FS90] Uriel Feige and Adi Shamir. Zero knowledge proofs of knowledge in two rounds. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 526–544. Springer, Heidelberg, August 1990.

[GHKW17] Rishab Goyal, Susan Hohenberger, Venkata Koppula, and Brent Waters. A generic approach to constructing and proving verifiable random functions. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017, Part II*, volume 10678 of *LNCS*, pages 537–566. Springer, Heidelberg, November 2017.

[GK96] Oded Goldreich and Ariel Kahan. How to construct constant-round zero-knowledge proof systems for NP. *Journal of Cryptology*, 9(3):167–190, June 1996.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.

[GMW91] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity for all languages in NP have zero-knowledge proof systems. *J. ACM*, 38(3):691–729, 1991.

[Gol01] Oded Goldreich. *The Foundations of Cryptography - Volume 1: Basic Techniques*. Cambridge University Press, 2001.

[Gol04] Oded Goldreich. *The Foundations of Cryptography - Volume 2: Basic Applications*. Cambridge University Press, 2004.

[Gra97] Jeroen Van De Graaf. *Towards a formal definition of security for quantum protocols*. PhD thesis, University of Montreal, Montreal, Canada, 1997.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.

[HM96] Shai Halevi and Silvio Micali. Practical and provably-secure commitment schemes from collision-free hashing. In Neal Koblitz, editor, *CRYPTO'96*, volume 1109 of *LNCS*, pages 201–215. Springer, Heidelberg, August 1996.

[JKKR17] Abhishek Jain, Yael Tauman Kalai, Dakshita Khurana, and Ron Rothblum. Distinguisher-dependent simulation in two rounds and its applications. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 158–189. Springer, Heidelberg, August 2017.

[Kob03] Hirotada Kobayashi. Non-interactive quantum perfect and statistical zero-knowledge. In Toshihide Ibaraki, Naoki Katoh, and Hirotaka Ono, editors, *ISAAC 2003*, volume 2906 of *Lecture Notes in Computer Science*, pages 178–188. Springer, 2003.

[LS19] Alex Lombardi and Luke Schaeffer. A note on key agreement and non-interactive commitments. Cryptology ePrint Archive, Report 2019/279, 2019. https://eprint.iacr.org/2019/279.

[Mah18a] Urmila Mahadev. Classical homomorphic encryption for quantum circuits. In Mikkel Thorup, editor, *59th FOCS*, pages 332–338. IEEE Computer Society Press, October 2018.

[Mah18b] Urmila Mahadev. Classical verification of quantum computations. In Mikkel Thorup, editor, *59th FOCS*, pages 259–267. IEEE Computer Society Press, October 2018.

[Nao91] Moni Naor. Bit commitment using pseudorandomness. *Journal of Cryptology*, 4(2):151–158, January 1991.

[NWZ09] Daniel Nagaj, Pawel Wocjan, and Yong Zhang. Fast amplification of qma. *arXiv*, 0904.1549, 2009.

[Pas03] Rafael Pass. Simulation in quasi-polynomial time, and its application to protocol composition. In Eli Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 160–176. Springer, Heidelberg, May 2003.

[Pei09] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In Michael Mitzenmacher, editor, *41st ACM STOC*, pages 333–342. ACM Press, May / June 2009.

[PS19] Chris Peikert and Sina Shiehian. Noninteractive zero knowledge for NP from (plain) learning with errors. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 89–114. Springer, Heidelberg, August 2019.

[PW08] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 187–196. ACM Press, May 2008.

[PW09] Rafael Pass and Hoeteck Wee. Black-box constructions of two-party protocols from one-way functions. In Omer Reingold, editor, *TCC 2009*, volume 5444 of *LNCS*, pages 403–418. Springer, Heidelberg, March 2009.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6):34:1–34:40, 2009.

[Shm20] Omri Shmueli. Multi-theorem (malicious) designated-verifier nizk for qma. *arXiv*, 2007.12923, 2020.

[SV03] Amit Sahai and Salil P. Vadhan. A complete problem for statistical zero knowledge. *J. ACM*, 50(2):196–249, 2003.

[Unr12] Dominique Unruh. Quantum proofs of knowledge. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 135–152. Springer, Heidelberg, April 2012.

[Unr16a] Dominique Unruh. Collapse-binding quantum commitments without random oracles. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 166–195. Springer, Heidelberg, December 2016.

[Unr16b] Dominique Unruh. Computationally binding quantum commitments. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 497–527. Springer, Heidelberg, May 2016.

[Wat09] John Watrous. Zero-knowledge against quantum attacks. *SIAM J. Comput.*, 39(1):25–58, 2009.

[Zha19] Mark Zhandry. Quantum lightning never strikes the same state twice. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part III*, volume 11478 of *LNCS*, pages 408–438. Springer, Heidelberg, May 2019.

A Omitted Preliminaries

We define post-quantum one-way functions and collapsing hash functions.

Definition A.1 (Post-Quantum One-Way Functions.)**.** *We say that a function $f : \{0,1\}^* \to \{0,1\}^*$ is post-quantum one-way function if f is computable in classical polynomial time and for any non-uniform QPT adversary $\mathcal{A}$, we have*

$$\Pr[f(x) = f(x') : x \xleftarrow{\$} \{0,1\}^\lambda, x' \xleftarrow{\$} \mathcal{A}(f(x))] = \mathsf{negl}(\lambda).$$

Definition A.2 (Collapsing Hash Function.)**.** *A length-decreasing function family $\mathcal{H} = \{H_k : \{0,1\}^L \to \{0,1\}^\ell\}_{k\in\mathcal{K}}$ for $L > \ell$ is collapsing if the following is satisfied:*

Collapsing. *For an adversary $\mathcal{A}$, we define an experiment $\mathsf{Exp}^{\mathsf{collapse}}_{\mathcal{A}}(1^\lambda)$ as follows:*

1. *The challenger generates $k \xleftarrow{\$} \mathcal{K}$.*
2. *$\mathcal{A}$ is given k as input and generates a hash value $y \in \{0,1\}^\ell$ and a quantum state σ over registers $(\mathbf{X}, \mathbf{A})$ where $\mathbf{X}$ stores an element of $\{0,1\}^L$ and $\mathbf{A}$ is $\mathcal{A}$'s internal register. Then it sends y and register $\mathbf{X}$ to the challenger, and keeps $\mathbf{A}$ on its side.*
3. *The challenger picks $b \xleftarrow{\$} \{0,1\}$ If $b = 0$, the challenger does nothing and if $b = 1$, the challenger measures register $\mathbf{X}$ in the computational basis. The challenger returns register $\mathbf{X}$ to $\mathcal{A}$*
4. *$\mathcal{A}$ outputs a bit b'. The experiment outputs 1 if $b' = b$ and 0 otherwise.*

We say that $\mathcal{A}$ is a valid adversary if we we have

$$\Pr[H_k(x) = y : k \xleftarrow{\$} \mathcal{K}, (y,\sigma) \xleftarrow{\$} \mathcal{A}(k), x \leftarrow M_{\mathbf{X}} \circ \sigma] = 1.$$

We say that a hash function is collapsing if for any non-uniform QPT valid adversary $\mathcal{A}$ we have

$$|\Pr[1 \xleftarrow{\$} \mathsf{Exp}^{\mathsf{collapsing}}_{\mathcal{A}}(1^\lambda)] - 1/2| = \mathsf{negl}(\lambda).$$

As shown in [Unr16a], a collapsing hash function with arbitrarily long (or even unbounded) input-length exists under the QLWE assumption.

B Construction of Strong Collapse-Binding Commitments

In this section, we show that (bounded-length) Halevi-Micali commitments [HM96, Unr16b] satisfies the strong collapse-binding property if we instantiate it based on a collapsing hash function.

Construction. In the following, we give a description of (bounded-length) Halevi-Micali commitments. Let $\mathcal{H} = \{H_k : \{0,1\}^L \to \{0,1\}^\ell\}_{k\in\mathcal{K}}$ be a family of collapsing hash functions and $\mathcal{F}$ be a family of universal hash functions $f : \{0,1\}^L \to \{0,1\}^n$ where $L = 4\ell + 2n + 4$. Then Halevi-Micali commitments over message space $\{0,1\}^n$ is described as follows:

$\mathsf{Setup}(1^\lambda)$**:** It chooses $k \xleftarrow{\$} \mathcal{K}$ and outputs $\mathsf{pp} := k$.

$\mathsf{Commit}(\mathsf{pp} = k, m)$**:** It picks $f \xleftarrow{\$} \mathcal{F}$ and $r \xleftarrow{\$} \{0,1\}^L$ conditioned on that $f(r) = m$, computes $y := H_k(r)$, and outputs $\mathsf{com} := (y, f)$.

This completes the description of the scheme. In the following, we prove security.

Statistical Hiding. This proof is completely identical to that in [HM96].

Strong Collapse-Binding. Suppose that there exists a non-uniform QPT adversary $\mathcal{A}$ that breaks the strong collapse-binding property of the above construction. Then we construct non-uniform QPT $\mathcal{B}$ that breaks the collapsing property of $\mathcal{H}$ as follows:

$\mathcal{B}(k)$**:** Given $k \in \mathcal{K}$, it sets $\mathsf{pp} := k$, sends pp to $\mathcal{A}$ as input, and receives $(\mathsf{com} = (y, f), \mathbf{M}, \mathbf{R})$ from $\mathcal{A}$. Then $\mathcal{B}$ sends $(y, \mathbf{R})$ to its external challenger where $\mathbf{R}$ plays the role of $\mathbf{X}$ in the collapsing game. Then it receives the register $\mathbf{R}$ (which is either measured or not) returned from the challenger and sends $(\mathbf{M}, \mathbf{R})$ to $\mathcal{A}$ as a response from the challenger. Finally, when $\mathcal{A}$ outputs b', then $\mathcal{B}$ also outputs b'.

First, if $\mathcal{A}$ is valid, then $\mathcal{B}$ is also valid since if (m, r) is a valid opening for $\mathsf{com} = (y, f)$, then we have $H_k(r) = y$. Moreover, since $(\mathbf{M}, \mathbf{R})$ contains a superposition of valid openings to com and the value of r completely determines m for a valid opening (m, r), measuring register $\mathbf{R}$ is equivalent to measuring both registers $(\mathbf{M}, \mathbf{R})$. Based on this observation, we can see that cases of $b = 0$ and $b = 1$ for the strong collapse-binding and collapsing games perfectly match. Therefore $\mathcal{B}$ breaks the collapsing with the same advantage as $\mathcal{A}$ breaks the strong collapse-binding.

Remark 7. *By a similar proof to that in [Unr16b, Unr16a], we can also prove the strong collapse-binding property for the unbounded-length version. We omit this since this is not needed for our purpose.*

C Equivalence between Definitions of Zero-Knowledge

Here, we introduce a seemingly stronger definition of quantum black-box ϵ-zero-knowledge than Definition 2.9, which captures entanglement between auxiliary input of a verifier and a distinguisher. Then we show that they are actually equivalent.

We define a seemingly stronger definition which we call *quantum black-box ϵ-zero-knowledge with entanglement*.

Definition C.1 (Post-Quantum Black-Box ϵ-Zero-Knowledge with Entanglement)**.** *For an interactive proof or argument for L, we define the following property:*

Quantum Black-Box ϵ-Zero-Knowledge with Entanglement. *There exists an oracle-aided QPT simulator* Sim *that satisfies the following: For any sequences of polynomial-size quantum circuits (referred to as malicious verifiers) $V^* = \{V^*_\lambda\}_{\lambda\in\mathbb{N}}$ that takes x as input and a state in a register $\mathbf{V}$ as advice and any noticeable function $\epsilon(\lambda)$, we define quantum channels $\Psi^{V^*}_{\mathsf{real},\lambda,x,w}$ and $\Psi^{V^*}_{\mathsf{Sim},\lambda,x}$ as follows:*

$\Psi^{V^*}_{\mathsf{real},\lambda,x,w}$**:** *Take a state σ in the register $\mathbf{V}$ as input and output $\mathsf{OUT}_{V^*_\lambda}\langle P(w), V^*_\lambda(\sigma)\rangle(x)$.*

$\Psi^{V^*}_{\mathsf{Sim},\lambda,x}$**:** *Take a state σ in the register $\mathbf{V}$ as input and output $\mathsf{OUT}_{V^*_\lambda}(\mathsf{Sim}^{V^*_\lambda(\sigma)}(x, 1^{\epsilon^{-1}}))$.*

Then for any sequence of polynomial-size states $\{\rho_\lambda\}_{\lambda\in\mathbb{N}}$ in registers $\mathbf{V}$ and any additional register $\mathbf{R}$, we have

$$\{(\Psi^{V^*}_{\mathsf{real},\lambda,x,w}\otimes I_{\mathbf{R}})(\rho_\lambda)\}_{\lambda,x,w} \overset{comp}{\approx}_\epsilon \{(\Psi^{V^*}_{\mathsf{Sim},\lambda,x}\otimes I_{\mathbf{R}})(\rho_\lambda)\}_{\lambda,x,w}$$

where $\lambda\in\mathbb{N}$, $x\in L\cap\{0,1\}^\lambda$, $w\in R_L(\lambda)$.

Lemma C.2. *If an interactive proof or argument satisfies quantum black-box ϵ-zero-knowledge (Definition 2.9), then it also satisfies quantum black-box ϵ-zero-knowledge with entanglement (Definition C.1).*

Proof. Suppose that we have sequences $\{V^*_\lambda\}_{\lambda\in\mathbb{N}}$ and $\{\rho_\lambda\}_{\lambda\in\mathbb{N}}$ as in the definition of quantum ϵ-zero-knowledge with entanglement in Definition C.1. For each λ, we consider a modified circuit $\widetilde{V}^*_\lambda$ that works similarly to V^*_λ except that it also takes a state on the register $\mathbf{R}$ as part of its advice but does not touch $\mathbf{R}$ at all. Then clearly we have

$$(\Psi^{V^*}_{\mathsf{real},\lambda,x,w}\otimes I_{\mathbf{R}})(\rho_\lambda) = \mathsf{OUT}_{\widetilde{V}^*_\lambda}\langle P(w), \widetilde{V}^*_\lambda(\rho_\lambda)\rangle(x).$$

By quantum black-box ϵ-zero-knowledge (Definition 2.9), there is a QPT simulator Sim such that we have

$$\{\mathsf{OUT}_{\widetilde{V}^*_\lambda}\langle P(w), \widetilde{V}^*_\lambda(\rho_\lambda)\rangle(x)\}_{\lambda,x,w} \overset{comp}{\approx}_\epsilon \{\mathsf{OUT}_{\widetilde{V}^*_\lambda}(\mathsf{Sim}^{\widetilde{V}^*_\lambda(\rho_\lambda)}(x, 1^{\epsilon^{-1}}))\}_{\lambda,x,w}$$

where $\lambda\in\mathbb{N}$, $x\in L\cap\{0,1\}^\lambda$, $w\in R_L(\lambda)$. By the definition of $\widetilde{V}^*_\lambda(\rho_\lambda)$, it does not act on register $\mathbf{R}$ and thus $\mathsf{Sim}^{\widetilde{V}^*_\lambda(\rho_\lambda)}$ does not act on $\mathbf{R}$ either. Therefore, we have

$$\mathsf{OUT}_{\widetilde{V}^*_\lambda}(\mathsf{Sim}^{\widetilde{V}^*_\lambda(\rho_\lambda)}(x, 1^{\epsilon^{-1}})) = (\mathsf{Sim}^{V^*_\lambda(\cdot)}(x, 1^{\epsilon^{-1}})\otimes I_{\mathbf{R}})(\rho_\lambda).$$

By combining the above, we can conclude that the protocol also satisfies the quantum ϵ-zero-knowledge with entanglement w.r.t. the same simulator Sim. □

Remark 8. *In the above proof, we do not use the full power of black-box simulation. Indeed, it suffices to assume a very mild condition that a simulator does not act on any register on which the verifier does not act. We also remark that the same proof works for equivalence between quantum black-box zero-knowledge and quantum black-box zero-knowledge with entanglement (which can be defined analogously).*

D Proof of Lemma 3.3

For proving Lemma 3.3, we first introduce the following lemma taken from [NWZ09], which is an easy consequence of Jordan's lemma.

Lemma D.1 ([NWZ09, Section 2.1])**.** *Let Π_0 and Π_1 be projectors on an N-dimensional Hilbert space $\mathcal{H}$. Then there is an orthogonal decomposition of $\mathcal{H}$ into two-dimensional subspaces S_j for $j \in [\ell]$ and $N - 2\ell$ one-dimensional subspaces $T_j^{(bc)}$ for $b, c \in \{0,1\}$ that satisfies the following properties:*

1. *For each two-dimensional subspace S_j, there exist two orthonormal bases $(|\alpha_j\rangle, |\alpha_j^\perp\rangle)$ and $(|\beta_j\rangle, |\beta_j^\perp\rangle)$ of S_j such that*

$$\Pi_0 |\alpha_j\rangle = |\alpha_j\rangle, \quad \Pi_0 |\alpha_j^\perp\rangle = 0,$$
$$\Pi_1 |\beta_j\rangle = |\beta_j\rangle, \quad \Pi_1 |\beta_j^\perp\rangle = 0.$$

   Moreover, if we let

$$p_j := \langle\alpha_j| \Pi_1 |\alpha_j\rangle,$$

   then we have $0 < p_i < 1$ and

$$|\alpha_j\rangle = \sqrt{p_j} |\beta_j\rangle + \sqrt{1-p_j} |\beta_j^\perp\rangle, |\beta_j\rangle = \sqrt{p_j} |\alpha_j\rangle + \sqrt{1-p_j} |\alpha_j^\perp\rangle.$$

2. *Each one-dimensional subspace $T_j^{(bc)}$ is spanned by a unit vector $|\alpha_j^{(bc)}\rangle$ such that $\Pi_0 |\alpha_j^{(bc)}\rangle = b |\alpha_j^{(bc)}\rangle$ and $\Pi_1 |\alpha_j^{(bc)}\rangle = c |\alpha_j^{(bc)}\rangle$.*

Then we prove Lemma 3.3.

Proof. (of Lemma 3.3.) We define projections Π_0 and Π_1 over $\mathcal{H} = \mathcal{H}_{\mathbf{X}} \times \mathcal{H}_{\mathbf{Y}}$ as

$$\Pi_0 := I_{\mathbf{X}} \otimes (|0\rangle\langle 0|)_{\mathbf{Y}}, \Pi_1 := \Pi,$$

and apply Lemma D.1 for them. In the following, we use the notations in Lemma D.1 for this particular application. We define

$$S_{<t} := \left(\bigoplus_{j:p_j<t} S_j\right) \oplus \left(\bigoplus_{j,b} T_j^{(b0)}\right)$$

and

$$S_{\geq t} := \left(\bigoplus_{j:p_j\geq t} S_j\right) \oplus \left(\bigoplus_{j,b} T_j^{(b1)}\right).$$

Then it is easy to see that they are an orthogonal decomposition of $\mathcal{H}$. Since each subspace is invariant under $\Pi_0 = I_{\mathbf{X}} \otimes (|0\rangle\langle 0|)_{\mathbf{Y}}$ and $\Pi_1 = \Pi$ by the Item 1 of Lemma D.1, Item 1 of Lemma 3.3 immediately follows.

For any quantum state $|\phi\rangle_{\mathbf{X}}$ such that $|\phi\rangle_{\mathbf{X}} |0\rangle_{\mathbf{Y}} \in S_{<t}$, we can write

$$|\phi\rangle_{\mathbf{X}} |0\rangle_{\mathbf{Y}} = \sum_{j:p_j<t} d_j |\alpha_j\rangle + \sum_j d_j^{(10)} |\alpha_j^{(10)}\rangle$$

by using $d_j \in \mathbb{C}$ and $d_j^{(10)} \in \mathbb{C}$ for each j such that $\sum_{j:p_j<t} |d_j|^2 + \sum_j |d_j^{(10)}|^2 = 1$.

Then we have

$$\begin{aligned}\langle\phi|_{\mathbf{X}} \langle 0|_{\mathbf{Y}} \Pi |\phi\rangle_{\mathbf{X}} |0\rangle_{\mathbf{Y}} &= \sum_{j:p_j<t} |d_j|^2 \langle\alpha_j| \Pi |\alpha_j\rangle + \sum_j |d_j^{(10)}|^2 \langle\alpha_j^{(10)}| \Pi |\alpha_j^{(10)}\rangle \\ &= \sum_{j:p_j<t} |d_j|^2 p_j \\ &< t.\end{aligned}$$

Similarly, for any $|\phi\rangle_{\mathbf{X}}$ such that $|\phi\rangle_{\mathbf{X}} |0\rangle_{\mathbf{Y}} \in S_{\geq t}$, we can show

$$\langle\phi|_{\mathbf{X}} \langle 0|_{\mathbf{Y}} \Pi |\phi\rangle_{\mathbf{X}} |0\rangle_{\mathbf{Y}} \geq t.$$

This completes the proof of Item 2 of Lemma 3.3.

For proving the Item 3 and 4 of Lemma 3.3, we first consider an algorithm $\widetilde{\mathsf{Amp}}$ described as follows:

$\widetilde{\mathsf{Amp}}(1^T, |\psi\rangle_{\mathbf{X},\mathbf{Y}})$**:** This algorithm takes a repetition parameter T and a quantum state $|\psi\rangle_{\mathbf{X},\mathbf{Y}} \in \mathcal{H}$ as input and works as follows:[26]

1. Repeat the following T times:
   (a) Perform a measurement $\{\Pi_1, I_{\mathbf{X},\mathbf{Y}} - \Pi_1\}$ If the outcome is 1, i.e., if Π_1 is applied, then output the state in the registers $(\mathbf{X}, \mathbf{Y})$ and a classical bit $b = 1$ indicating a success and immediately halt.
   (b) Perform a measurement $\{\Pi_0, I_{\mathbf{X},\mathbf{Y}} - \Pi_0\}$.
2. Output the state in the registers $(\mathbf{X}, \mathbf{Y})$ and a classical bit $b = 0$ indicating a failure.

Then we prove the following claim:

Claim D.2. *The following hold*

1. *For any quantum state $|\psi\rangle_{\mathbf{X},\mathbf{Y}}$, if we run $(|\psi'\rangle_{\mathbf{X},\mathbf{Y}}, b) \xleftarrow{\$} \widetilde{\mathsf{Amp}}(1^T, |\psi\rangle_{\mathbf{X},\mathbf{Y}})$ and we have $b = 1$, then $|\psi'\rangle_{\mathbf{X},\mathbf{Y}}$ is in the span of Π_1 with probability 1.*
2. *For any noticeable function $\nu = \nu(\lambda)$, there is $T = \mathsf{poly}(\lambda)$ such that for any quantum state $|\phi\rangle_{\mathbf{X}}$ such that $|\phi\rangle_{\mathbf{X}} |0\rangle_{\mathbf{Y}} \in S_{\geq t}$, we have*
$$\Pr[b = 1 : (|\psi'\rangle_{\mathbf{X},\mathbf{Y}}, b) \xleftarrow{\$} \widetilde{\mathsf{Amp}}(1^T, |\phi\rangle_{\mathbf{X}} |0\rangle_{\mathbf{Y}})] \geq 1 - \nu.$$
3. *$S_{<t}$ and $S_{\geq t}$ are invariant under $\widetilde{\mathsf{Amp}}(1^T, \cdot)$. More precisely, for any quantum state $|\psi_{<t}\rangle_{\mathbf{X},\mathbf{Y}} \in S_{<t}$ if we run $(|\psi'\rangle_{\mathbf{X},\mathbf{Y}}, b) \xleftarrow{\$} \widetilde{\mathsf{Amp}}(1^T, |\psi_{<t}\rangle_{\mathbf{X},\mathbf{Y}})$, then we have $|\psi'\rangle_{\mathbf{X},\mathbf{Y}} \in S_{<t}$ with probability 1. Similarly, for any quantum state $|\psi_{\geq t}\rangle_{\mathbf{X},\mathbf{Y}} \in S_{\geq t}$ if we run $(|\psi'\rangle_{\mathbf{X},\mathbf{Y}}, b) \xleftarrow{\$} \widetilde{\mathsf{Amp}}(1^T, |\psi_{\geq t}\rangle_{\mathbf{X},\mathbf{Y}})$, then we have $|\psi'\rangle_{\mathbf{X},\mathbf{Y}} \in S_{\geq t}$ with probability 1.*

[26] Strictly speaking, we need to consider descriptions of quantum circuits to perform measurements $\{\Pi_0, I_{\mathbf{X},\mathbf{Y}} - \Pi_0\}$ and $\{\Pi_1, I_{\mathbf{X},\mathbf{Y}} - \Pi_1\}$ as part of its input so that we can make the description of $\widetilde{\mathsf{Amp}}$ independent on them. (Looking ahead, this is needed for showing Item 4 in Lemma 3.3 where Amp is required to be a uniform QPT machine.) We omit to explicitly write them in the input of $\widetilde{\mathsf{Amp}}$ for notational simplicity.

Proof. (of Claim D.2.) Item 1 immediately follows from the description of $\widetilde{\mathsf{Amp}}$ since it returns $b=1$ only after succeeding in applying Π_1. Item 3 is easy to see noting that $\widetilde{\mathsf{Amp}}$ just sequentially applies measurements $\{\Pi_1, I_{\mathbf{X},\mathbf{Y}}-\Pi_1\}$ and $\{\Pi_0, I_{\mathbf{X},\mathbf{Y}}-\Pi_0\}$ over the registers $\mathbf{X}$ and $\mathbf{Y}$ and each subspace S_j or $T_j^{(bc)}$ is invariant under these measurements by Lemma D.1. In the following, we prove Item 2. We note that essentially the same statement was proven in [CCY20]. We include a proof for completeness.

For any quantum state $|\phi\rangle_{\mathbf{X}}$ such that $|\phi\rangle_{\mathbf{X}}|0\rangle_{\mathbf{Y}} \in S_{\geq t}$, we can write

$$|\phi\rangle_{\mathbf{X}}|0\rangle_{\mathbf{Y}} = \sum_{j:p_j\geq t} d_j|\alpha_j\rangle + \sum_j d_j^{(11)}|\alpha_j^{(11)}\rangle$$

by using $d_j \in \mathbb{C}$ and $d_j^{(11)} \in \mathbb{C}$ for each j such that $\sum_{j:p_j\geq t}|d_j|^2 + \sum_j |d_j^{(11)}|^2 = 1$. Since $\widetilde{\mathsf{Amp}}$ just sequentially applies measurements $\{\Pi_1, I_{\mathbf{X},\mathbf{Y}}-\Pi_1\}$ and $\{\Pi_0, I_{\mathbf{X},\mathbf{Y}}-\Pi_0\}$ over the registers $\mathbf{X}$ and $\mathbf{Y}$ and each subspace S_j or $T_j^{(bc)}$ is invariant under these measurements by Lemma D.1, states in different subspaces do not interfere with each other. Therefore, it suffices to prove Item 2 of D.2 assuming that $|\phi\rangle_{\mathbf{X}}|0\rangle_{\mathbf{Y}} = |\alpha_j\rangle$ for some j such that $p_j \geq t$ or $|\phi\rangle_{\mathbf{X}}|0\rangle_{\mathbf{Y}} = |\alpha_j^{(11)}\rangle$ for some j.

The latter case is easy: If $|\phi\rangle_{\mathbf{X}}|0\rangle_{\mathbf{Y}} = |\alpha_j^{(11)}\rangle$ for some j, then $\widetilde{\mathsf{Amp}}(1^T, |\alpha_j^{(11)}\rangle)$ outputs $(|\alpha_j^{(11)}\rangle, b=1)$ and halts at the very first step with probability 1 since we have $\Pi_1|\alpha_j^{(11)}\rangle = |\alpha_j^{(11)}\rangle$ by Lemma D.1.

In the following, we analyze the case of $|\phi\rangle_{\mathbf{X}}|0\rangle_{\mathbf{Y}} = |\alpha_j\rangle$ for some j such that $p_j \geq t$. For $k\in\mathbb{N}$, let P_k and $P_k^\perp$ be the probability that $\widetilde{\mathsf{Amp}}(1^k, |\alpha_j\rangle)$ and $\widetilde{\mathsf{Amp}}(1^k, |\alpha_j^\perp\rangle)$ succeed, respectively. (We define $P_0 = P_0^\perp := 0$.) By using

$$|\alpha_j\rangle = \sqrt{p_j}|\beta_j\rangle + \sqrt{1-p_j}|\beta_j^\perp\rangle, |\beta_j\rangle = \sqrt{p_j}|\alpha_j\rangle + \sqrt{1-p_j}|\alpha_j^\perp\rangle,$$

we can see that we have

$$\begin{aligned} P_{k+1} &= p_j + (1-p_j)^2 P_k + (1-p_j)p_j P_k^\perp, \\ P_{k+1}^\perp &= (1-p_j) + p_j(1-p_j)P_k + p_j^2 P_k^\perp. \end{aligned}$$

Solving this, we have

$$P_T = 1 - (1-2p_j+2p_j^2)^{T-1}(1-p_j)$$

for $T \geq 1$. Since $p_j \geq t$ and ν are noticeable, we can take $T = \mathsf{poly}(\lambda)$ in such a way that $P_T \geq 1-\nu$. This completes the proof of Claim D.2. □

We define an algorithm Amp as a purified version of $\widetilde{\mathsf{Amp}}$. That is, Amp works similarly to $\widetilde{\mathsf{Amp}}$ except that intermediate measurement results are stored in designated registers in $\mathbf{Anc}$ without being measured and the output b is stored in register $\mathbf{B}$. Let $U_{\mathsf{amp},T}$ be the unitary part of $\mathsf{Amp}(1^T,\cdot)$. Then Item 3 of Lemma 3.3 directly follows from the corresponding statements of Claim D.2. Finally, Amp clearly runs in QPT by the definition, and thus Item 4 of Lemma 3.3 follows. □

E Proof of Lemma 5.5

Here, we give a proof of Lemma 5.5.

Proof. (of Lemma 5.5). We consider the following probabilities:

$p^{\mathsf{suc}}_{\mathsf{real}}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$**:** A probability that V^*_λ does not abort in an execution $\langle P(w), V^*_\lambda(\rho_\lambda)\rangle(x)$.

$p^{\mathsf{suc}}_{\mathsf{a}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$**:** A probability that $\mathsf{Sim}_{\mathsf{a}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$ does not return Fail.

$p^{\mathsf{suc}}_{\mathsf{na}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$**:** A probability that $\mathsf{Sim}_{\mathsf{na}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$ does not return Fail.

Lemma 5.3 and 5.4 immediately imply that there is a negligible function negl such that for all $x = \{x_\lambda \in L \cap \{0,1\}^\lambda\}_{\lambda\in\mathbb{N}}$ we have

$$p^{\mathsf{suc}}_{\mathsf{a}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda) = p^{\mathsf{suc}}_{\mathsf{real}}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda) \tag{12}$$

and

$$\left|p^{\mathsf{suc}}_{\mathsf{na}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda) - (1 - p^{\mathsf{suc}}_{\mathsf{real}}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda))\right| \leq \delta + \mathsf{negl}(\lambda). \tag{13}$$

By the construction of $\mathsf{Sim}_{\mathsf{comb}}(x, V^*_\lambda, \rho_\lambda)$, we have

$$p^{\mathsf{suc}}_{\mathsf{comb}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda) = \frac{1}{2}\left(p^{\mathsf{suc}}_{\mathsf{a}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda) + p^{\mathsf{suc}}_{\mathsf{na}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)\right). \tag{14}$$

Combining Eq. 12, 13, and 14, we obtain Eq. 8.

For proving Eq. 9, We consider the following distributions:

$D_{\mathsf{real,a}}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$: A conditional distribution of $\mathsf{OUT}_{V^*_\lambda}\langle P(w), V^*_\lambda(\rho_\lambda)\rangle(x)$, conditioned on that V^*_λ aborts.

$D_{\mathsf{real,na}}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$: A conditional distribution of $\mathsf{OUT}_{V^*_\lambda}\langle P(w), V^*_\lambda(\rho_\lambda)\rangle(x)$, conditioned on that V^*_λ does not abort.

$D_{\mathsf{sim,a}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$: A conditional distribution of $\mathsf{Sim}_{\mathsf{a}}(x, V^*_\lambda, \rho_\lambda)$, conditioned on that the output is not Fail.

$D_{\mathsf{sim,na}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$: A conditional distribution of $\mathsf{Sim}_{\mathsf{na}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$, conditioned on that the output is not Fail.

Then we consider the following sequence of distributions implicitly indexed by $\lambda \in \mathbb{N}$, $x \in L \cap \{0,1\}^\lambda$, and $w \in R_L(x)$.

$D_1 := D_{\mathsf{sim,comb}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$. We note that this can be rephrased as follows:

It samples from $D_{\mathsf{sim,a}}(x, V^*_\lambda, \rho_\lambda)$ with probability

$$\frac{p^{\mathsf{suc}}_{\mathsf{a}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)}{p^{\mathsf{suc}}_{\mathsf{a}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda) + p^{\mathsf{suc}}_{\mathsf{na}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)}$$

and from $D_{\mathsf{sim,na}}(x, V^*_\lambda, \rho_\lambda)$ with probability

$$\frac{p^{\mathsf{suc}}_{\mathsf{na}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)}{p^{\mathsf{suc}}_{\mathsf{a}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda) + p^{\mathsf{suc}}_{\mathsf{na}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)}.$$

D_2 : It samples from $D_{\mathsf{sim,a}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$ with probability $p^{\mathsf{suc}}_{\mathsf{real}}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$ and from $D_{\mathsf{sim,na}}(x, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$ with probability $1 - p^{\mathsf{suc}}_{\mathsf{real}}(x, w, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$.

D_3 : It samples from $D_{\mathsf{real},\mathsf{a}}(x,w,1^{\epsilon^{-1}},V^*_\lambda,\rho_\lambda)$ with probability $p^{\mathsf{suc}}_{\mathsf{real}}(x,w,1^{\epsilon^{-1}},V^*_\lambda,\rho_\lambda)$ and from $D_{\mathsf{sim},\mathsf{na}}(x,1^{\epsilon^{-1}},V^*_\lambda,\rho_\lambda)$ with probability $1-p^{\mathsf{suc}}_{\mathsf{real}}(x,w,1^{\epsilon^{-1}},V^*_\lambda,\rho_\lambda)$.

D_4 : It samples from $D_{\mathsf{real},\mathsf{a}}(x,w,1^{\epsilon^{-1}},V^*_\lambda,\rho_\lambda)$ with probability $1-p^{\mathsf{suc}}_{\mathsf{na}}(x,1^{\epsilon^{-1}},V^*_\lambda,\rho_\lambda)$ and from $D_{\mathsf{real},\mathsf{na}}(x,1^{\epsilon^{-1}},V^*_\lambda,\rho_\lambda)$ with probability $p^{\mathsf{suc}}_{\mathsf{na}}(x,1^{\epsilon^{-1}},V^*_\lambda,\rho_\lambda)$.

D_5 : It samples from $D_{\mathsf{real},\mathsf{a}}(x,w,1^{\epsilon^{-1}},V^*_\lambda,\rho_\lambda)$ with probability $p^{\mathsf{suc}}_{\mathsf{real}}(x,w,1^{\epsilon^{-1}},V^*_\lambda,\rho_\lambda)$ and from $D_{\mathsf{real},\mathsf{na}}(x,1^{\epsilon^{-1}},V^*_\lambda,\rho_\lambda)$ with probability $1-p^{\mathsf{suc}}_{\mathsf{real}}(x,w,1^{\epsilon^{-1}},V^*_\lambda,\rho_\lambda)$.
We can see that this is exactly equal to $\mathsf{OUT}_{V^*_\lambda}\langle P(w),V^*_\lambda(\rho_\lambda)\rangle(x)$

In the following, we give an upper bound for advantage to distinguish each neighboring distributions. We denote by $\mathcal{D}_j$ to mean $\{D_j\}_{\lambda\in\mathbb{N},x\in L\cap\{0,1\}^\lambda,w\in R_L(x)}$.

- $\mathcal{D}_1 \overset{stat}{\approx}_{2\delta} \mathcal{D}_2$: By Eq. 12 and 13, we have
$$1-\delta-\mathsf{negl}(\lambda)\le p^{\mathsf{suc}}_{\mathsf{a}}(x,1^{\epsilon^{-1}},V^*_\lambda,\rho_\lambda)+p^{\mathsf{suc}}_{\mathsf{na}}(x,1^{\epsilon^{-1}},V^*_\lambda,\rho_\lambda)\le 1+\delta+\mathsf{negl}(\lambda).$$
Then, by using Eq. 12 again, we have
$$\left|\frac{p^{\mathsf{suc}}_{\mathsf{a}}(x,1^{\epsilon^{-1}},V^*_\lambda,\rho_\lambda)}{p^{\mathsf{suc}}_{\mathsf{a}}(x,1^{\epsilon^{-1}},V^*_\lambda,\rho_\lambda)+p^{\mathsf{suc}}_{\mathsf{na}}(x,1^{\epsilon^{-1}},V^*_\lambda,\rho_\lambda)}-p^{\mathsf{suc}}_{\mathsf{real}}(x,w,1^{\epsilon^{-1}},V^*_\lambda,\rho_\lambda)\right|\le 2\delta+\mathsf{negl}(\lambda)$$
where we use $\delta<\epsilon=o(1)$ and in particular $\delta<1/2$ for a sufficiently large λ and $1-z<\frac{1}{1+z}$ and $\frac{1}{1-z}<1+2z$ for all reals $0<z<1/2$. Then $\mathcal{D}_1 \overset{stat}{\approx}_{2\delta} \mathcal{D}_2$ immediately follows.[27]
- $\mathcal{D}_2\equiv\mathcal{D}_3$: This immediately follows from Lemma 5.3 since it implies $D_{\mathsf{sim},\mathsf{a}}(x,V^*_\lambda,\rho_\lambda)$ and $D_{\mathsf{real},\mathsf{a}}(x,w,1^{\epsilon^{-1}},V^*_\lambda,\rho_\lambda)$ are exactly the same distributions.
- $\mathcal{D}_3 \overset{comp}{\approx}_{\delta} \mathcal{D}_4$: Here, we denote by $D_{j,\lambda,x,w}$ to mean D_j for clarifying the dependence on λ,x,w. We consider distributions D_{3,λ,x,w,ρ^*} and D_{4,λ,x,w,ρ^*} for any state ρ^* in the support of $D_{\mathsf{real},\mathsf{a}}(x,w,1^{\epsilon^{-1}},V^*_\lambda,\rho_\lambda)$ defined as follows:

  D_{3,λ,x,w,ρ^*} : It outputs ρ^* with probability $p^{\mathsf{suc}}_{\mathsf{real}}(x,w,1^{\epsilon^{-1}},V^*_\lambda,\rho_\lambda)$ and samples from $D_{\mathsf{sim},\mathsf{na}}(x,1^{\epsilon^{-1}},V^*_\lambda,\rho_\lambda)$ with probability $1-p^{\mathsf{suc}}_{\mathsf{real}}(x,w,1^{\epsilon^{-1}},V^*_\lambda,\rho_\lambda)$.

  D_{4,λ,x,w,ρ^*} : It outputs ρ^* with probability $1-p^{\mathsf{suc}}_{\mathsf{na}}(x,1^{\epsilon^{-1}},V^*_\lambda,\rho_\lambda)$ and samples from $D_{\mathsf{real},\mathsf{na}}(x,1^{\epsilon^{-1}},V^*_\lambda,\rho_\lambda)$ with probability $p^{\mathsf{suc}}_{\mathsf{na}}(x,1^{\epsilon^{-1}},V^*_\lambda,\rho_\lambda)$.

  Suppose that $\mathcal{D}_3 \overset{comp}{\approx}_{\delta} \mathcal{D}_4$ does not hold. This means that there exists a non-uniform PPT distinguisher $\mathcal{A}=\{\mathcal{A}_\lambda,\rho_{\mathcal{A},\lambda}\}_{\lambda\in\mathbb{N}}$ and a sequence $\{(x_\lambda,w_\lambda)\in(L\cap\{0,1\}^\lambda)\times R_L(x)\}_{\lambda\in\mathbb{N}}$ such that
$$|\Pr[1\xleftarrow{\$}\mathcal{A}_\lambda(D_{3,\lambda,x_\lambda,w_\lambda})]-\Pr[1\xleftarrow{\$}\mathcal{A}(D_{4,\lambda,x_\lambda,w_\lambda})]-\delta$$
is non-negligible. By an averaging argument, there exists a sequence $\{\rho^*_\lambda\}_{\lambda\in\mathbb{N}}$ such that
$$|\Pr[1\xleftarrow{\$}\mathcal{A}_\lambda(D_{3,\lambda,x_\lambda,w_\lambda,\rho^*_\lambda})]-\Pr[1\xleftarrow{\$}\mathcal{A}(D_{4,\lambda,x_\lambda,w_\lambda,\rho^*_\lambda})]-\delta$$
is non-negligible. We fix such $\{\rho^*_\lambda\}_{\lambda\in\mathbb{N}}$. By using $\mathcal{A}$, we construct a non-uniform QPT distinguisher $\mathcal{A}'$ that is given $\{\rho_{\mathcal{A},\lambda},\rho^*_\lambda\}_{\lambda\in\mathbb{N}}$ as an advice and distinguishes $\mathsf{OUT}_{V^*_{\mathsf{na}}}\langle P(w_\lambda),V^*_\lambda(\rho_\lambda)\rangle(x_\lambda)$ and $\mathsf{Sim}_{\mathsf{na}}(x_\lambda,1^{\epsilon^{-1}},V^*_\lambda,\rho_\lambda)$ as follows:

[27] Indeed, we can prove $\mathcal{D}_1 \overset{stat}{\approx}_{c\delta} \mathcal{D}_2$ for any constant $c>1$ similarly.

$\mathcal{A}'(\rho')$**:** It is given an input ρ', which is sampled from either $\mathsf{OUT}_{V^*_{\mathsf{na}}}\langle P(w_\lambda), V^*_\lambda(\rho_\lambda)\rangle(x_\lambda)$ or $\mathsf{Sim}_{\mathsf{na}}(x_\lambda, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda)$. It sets $\rho := \rho'$ if $\rho' \neq \mathsf{Fail}$ and otherwise sets $\rho := \rho^*_\lambda$. Then it runs $\mathcal{A}$ on input ρ and outputs as $\mathcal{A}$ outputs.

Clearly, if ρ' is sampled from $\mathsf{OUT}_{V^*_{\mathsf{na}}}\langle P(w_\lambda), V^*_\lambda(\rho_\lambda)\rangle(x_\lambda)$, then ρ is distributed according to $D_{3,x_\lambda,w_\lambda,\rho^*_\lambda}$ and if ρ' is sampled from $\mathsf{Sim}_{\mathsf{na}}(x_\lambda, V^*_\lambda, \rho_\lambda)$, then ρ is distributed according to $D_{4,x_\lambda,w_\lambda,\rho^*_\lambda}$. Therefore,

$$|\Pr[1 \xleftarrow{\$} \mathcal{A}'(\mathsf{OUT}_{V^*_{\mathsf{na}}}\langle P(w_\lambda), V^*_\lambda(\rho_\lambda)\rangle(x_\lambda))] - \Pr[1 \xleftarrow{\$} \mathcal{A}'(\mathsf{Sim}_{\mathsf{na}}(x_\lambda, 1^{\epsilon^{-1}}, V^*_\lambda, \rho_\lambda))]| - \delta$$

is non-negligible. This contradicts Lemma 5.4. Therefore, we have $\mathcal{D}_3 \overset{comp}{\approx}_\delta \mathcal{D}_4$.

- $\mathcal{D}_4 \overset{comp}{\approx}_\delta \mathcal{D}_5$: This immediately follows from Eq. 13.

By combining the above, we obtain Eq. 9. □

Contents

1 Introduction **1**
- 1.1 Our Results . . . 2
- 1.2 Technical Overview . . . 3
  - 1.2.1 Quantum Analysis of GK Protocol. . . . 4
  - 1.2.2 OWF-based Construction. . . . 8
- 1.3 Related Work . . . 9

2 Preliminaries **11**
- 2.1 Post-Quantum One-Way Functions and Collapsing Hash Functions . . . 12
- 2.2 Commitment . . . 12
- 2.3 Interactive Proof and Argument. . . . 15
  - 2.3.1 Witness Indistinguishable Proof of Knowledge . . . 15
  - 2.3.2 Delayed-Witness Σ-Protocol . . . 16
  - 2.3.3 Quantum ϵ-Zero-Knowledge Proof and Argument . . . 17
- 2.4 Quantum Rewinding Lemma . . . 18

3 Technical Lemmas **19**

4 Extraction Lemma **22**
- 4.1 Proof of Extraction Lemma (Lemma 4.2) . . . 23
- 4.2 Proof of Claim 4.5 . . . 25

5 Post-Quantum ϵ-Zero-Knowledge Proof **27**
- 5.1 Construction . . . 28
- 5.2 Statistical Soundness . . . 28
- 5.3 Quantum Black-Box ϵ-Zero-Knowledge . . . 29
- 5.4 Instantiation from Collapsing Hash Function . . . 34

6 Post-Quantum ϵ-Zero-Knowledge Argument from OWF **34**
- 6.1 Preparation . . . 34
- 6.2 Construction . . . 36
- 6.3 Computational Soundness . . . 36
- 6.4 Quantum Black-Box ϵ-Zero-Knowledge . . . 38

A Omitted Preliminaries **46**

B Construction of Strong Collapse-Binding Commitments **47**

C Equivalence between Definitions of Zero-Knowledge **47**

D Proof of Lemma 3.3 **49**

E Proof of Lemma 5.5 **51**